AD-A151 043 VISUAL RECOGNITION OF SIMPLE OBJECTS BY A CONNECTION NETWORK(U) ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE D C PLAUT AUG 84 TR-143 N00014-82-K-0193

1/1

UNCLASSIFIED

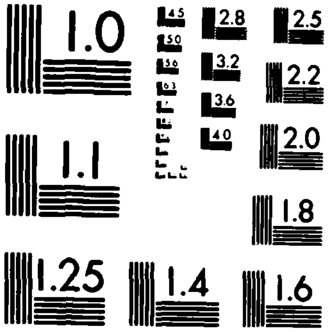
F/G 12/1

NL

END

FILMED

DTIC


1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

# Visual Recognition of Simple Objects by a Connection Network

David C. Plaut
Computer Science Department
University of Rochester
Rochester, NY 14627

TR143
August 1984

rochester

Department of Computer Science
University of Rochester
Rochester, New York 14627

# Visual Recognition of Simple Objects by a Connection Network

David C. Plaut
Computer Science Department
University of Rochester
Rochester, NY 14627

TR143
August 1984

## Abstract

A difficult problem in vision research is specifying how meaningful objects are recognized using the visual feature information extracted from an image. The fundamental issue involves the interaction of different levels of representation of visual information. The technical and theoretical problems that must be addressed in specifying this interaction arise in any attempt to model visual object perception. We attempt to deal with some difficult aspects of this process within the context of Feldman's Four Frames model of visual perception. ~~[Feldman, 1984]~~.

The model consists of four continually interacting representational frames, expressed in terms of a massively parallel, *connectionist* formalism. Within the Four Frames model, the problem of accessing object representations using visual feature information can be defined in specific computational terms. ~~We present in~~ this paper, the detailed design of a connectionist model as a possible solution to some of the major problems in the visual recognition of objects. The model proposes that an object is represented as a hierarchical structure of geometric subparts. Recognition proceeds by determining in parallel that all subparts of an object are present in the image, and then sequentially verifying that each subpart is in the proper spatial relation to the others. Implementation results demonstrate that the model can recognize any of a set of simple objects given fairly general feature input. Although the model is developed in the context of a drastically simplified visual domain, the principles it embodies are argued to adhere to many of the behavioral and biological constraints of real-world vision.

-1

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TR 143

2. GOVT ACCESSION NO.
AD-A151 043

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
Visual Recognition of Simple Objects by a Connection Network

5. TYPE OF REPORT & PERIOD COVERED
technical report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
David C. Plaut

8. CONTRACT OR GRANT NUMBER(s)
N00014-82-K-0193

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer Science Department
University of Rochester
Rochester, NY 14627

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
August 1984

13. NUMBER OF PAGES
26

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. *(of this report)*
-unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A difficult problem in vision research is specifying how meaningful objects are recognized using the visual feature information extracted from an image. The fundamental issue involves the interaction of different levels of representation of visual information. The technical and theoretical problems that must be addressed in specifying this interaction arise in any attempt to model visual object perception. We attempt to deal with some difficult aspects of this process within the context of Feldman's Four Frames model of visual perception (Feldman, 1984).

unclassified
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. Abstract (cont.)

The model consists of four continually interacting representational frames, expressed in terms of a massively parallel, connectionist formalism. Within the Four Frames model, the problem of accessing object representations using visual feature information can be defined in specific computational terms. We present in this paper the detailed design of a connectionist model as a possible solution to some of the major problems in the visual recognition of objects. The model proposes that an object is represented as a hierarchical structure of geometric subparts. Recognition proceeds by determining in parallel that all subparts of an object are present in the image, and then sequentially verifying that each subpart is in the proper spatial relation to the others. Implementation results demonstrate that the model can recognize any of a set of simple objects given fairly general feature input. Although the model is developed in the context of a drastically simplified visual domain, the principles it embodies are argued to adhere to many of the behavioral and biological constraints of real-world vision.

## 1. Introduction

The fundamental goal of vision research is to specify how information contained in light absorbed by the retinae is processed into meaningful information about the world. A central problem that must be addressed is how meaningful objects are identified from the overwhelming amount of low-level visual information that must be processed. The major difficulty stems from the large representational gap between image information and the object descriptions which explain this information. Bridging this gap seems to require the use of a number of interrelated intermediate levels of representation [Ballard & Brown, 1982].

Early work in object recognition involved extracting line drawings from images of polyhedral scenes, and interpreting them using geometrical object prototypes [Roberts, 1965; Falk, 1972]. The unreliability and inflexibility of these early approaches stemmed, in part, from the lack of concise and general representations to organize visual information at a lower level than objects [Barrow & Tenenbaum, 1975; Zucker *et al.*, 1975]. [Waltz, 1975] employed line junctions as an intermediate form of representation, and exploited the inherent constraints between the different types of possible junctions to successfully analyze complex polyhedral scenes. Unfortunately, these constraints critically depend on the specific nature of polyhedra and do not generalize to more natural domains. More recently, [Barrow & Tenenbaum, 1978] demonstrated the plausibility of deriving more general *intrinsic image* information (e.g. surface reflectance, distance, surface orientation, optical flow) from images for use as an intermediate representational level in object recognition. Yet converting from these intrinsic images to an object-based representation remains a difficult problem.

Much more work has been done in developing parsimonious object representations than in specifying their *use* in object recognition. Marr and Nishihara [Marr & Nishihara, 1978] have argued for geometrically-based hierarchical object representations. They employ generalized cones as the fundamental descriptive elements, although any set of primitives with a geometrical basis and defining coordinate frame is sufficient [Hrechanyk & Ballard, 1982]. [Marr, 1982] proposes that these representations are accessed using information analogous to intrinsic images (a "2 1/2 - D sketch" based on depth and surface orientation) but the technical details of this process have yet to be worked out.

Object recognition has come to be viewed within vision research as the process by which visual information in an image is matched with one of a collection of known object representations. A detailed computational account of this process must take place in the context of a particular model of vision, but the problems that arise must be addressed by any comprehensive model of visual perception. The purpose of this paper is to propose a mechanism for relating visual feature information to object representations in a way compatible with known biological and behavioral data.

The rest of this section describes the general model of vision which forms the basis for our detailed discussion of the problems involved in object recognition. In order to make specific computational claims about this process, we require a computational formalism suitable for expressing the highly parallel interaction of information required in vision. We present an overview of such a formalism in Section 2. Since our model must relate visual features to object representations, Section 3 addresses the nature of the representation of visual feature information and the initial problems in making the jump from a syntactic to a semantic representation. In Section 4 we elaborate on the nature of object representations and addfess some of the difficulties in mapping this object-centered information to the viewer-centered information in an image. Section 5 presents our detailed model of object recognition specified in terms of the computational formalism of Section 2. Also included in Section 5 are the results of an implementation of the model which demonstrate that it is capable of recognizing any of a set of simple objects given fairly general feature input. Finally, in Section 6 we describe how the model relates to some of the relevant behavioral and biological data on object perception, as well as suggest future directions for the development of the model.

## 1.1 Four Frames Model

[Feldman, 1984] presents what is argued to be a computationally sufficient and scientifically plausible model of how primates perceive objects and deal with their visual environments. The various visual processes involved are specified in terms of the operation and interaction of four computational frames. Within the model, a *frame* consists of a class of information and the processes operating on it which together share a particular representation. Each of the four frames processes a different type of information and all are assumed to be constantly active and interacting. The basic organization of the model is shown in Figure 1.1.

Figure 1.1: The Four Frames model

The *Retinotopic Frame* (RF) consists of the computational structures operating on information specific to each eye fixation. It receives input in the form of light intensity levels distributed over a two-dimensional surface, and performs local grouping and smoothing operations on this information. A central, foveal region is capable of the much higher resolution processing required for certain complex calculations such as color and texture.

The *Stable Feature Frame* (SFF) captures the visual information that is stable over eye fixations. This information can be thought roughly to be relative to a particular head position (i.e. it is *viewer-centered*), giving rise to the perception of a stable visual world. The SFF computes distal (constancy, intrinsic) feature values from the proximal stimulus features of the RF. The variations in processing associated with eye movements from a stable viewing position are captured by a *gaze* mapping between positions in the RF (particularly the fovea) and the feature computations at positions in the SFF. The exact nature of the features computed by the SFF is not critical to the model (as will be explained later), but the features are assumed to be similar to intrinsic images.

The third frame (the *World Knowledge Formulary*, or WKF) is unlike the first two in that it is not geometrical and not modality-specific. The WKF encodes the observer's general knowledge of the world as a particular kind of semantic network. Included as part of the WKF representation of an object is knowledge of the object's

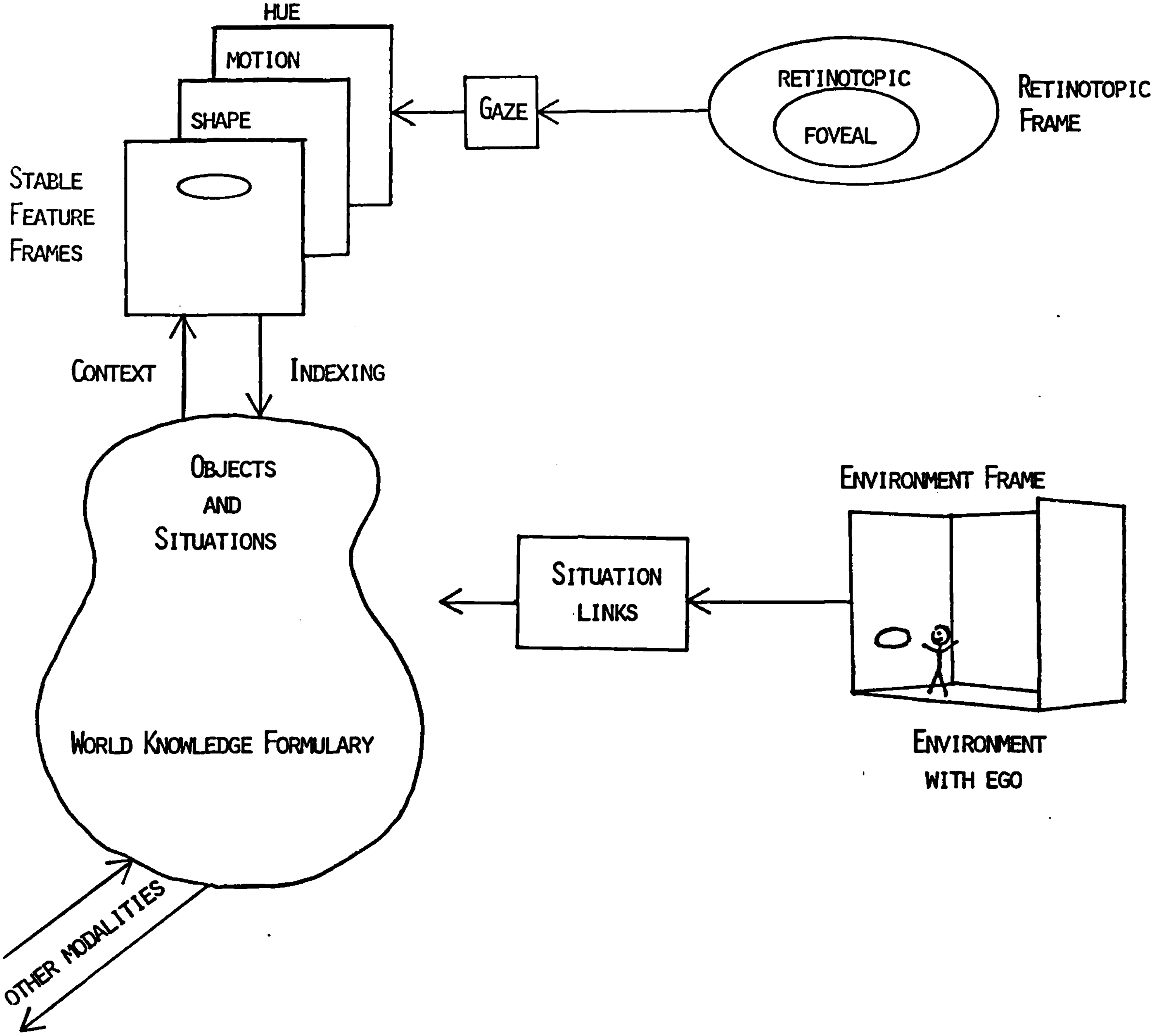


FIGURE 1.1: FOUR FRAMES, MAJOR STRUCTURES AND LINKS

visual appearance. Visual feature information in the SFF provides access to (indexes) this knowledge during recognition of the object. Conversely, knowledge and expectations in the WKF can provide top-down context for directing and enhancing the SFF feature processing used in object recognition.

The final, *Environmental Frame* (EF), models an observer's representation of the surrounding space. The representation is assumed to be allocentric (external) and separate computations continually update information on the observer's position within the environment. *Situation links* between the EF and WKF maintain information on what objects are in what positions in the current visual situation. Processing within the EF determines what positions are viewable from the observer's position, and what objects should be visible. These expectations direct attention mechanisms (using the RF-SFF *gaze* mapping) to focus processing on positions of interest, allowing the observer to more effectively gather visual information about the environment.

## 1.2 Recognition

Within the Four Frames model, the recognition problem can be precisely characterized in terms of the interaction between the SFF and WKF: visual features in the SFF must combine to access the meaningful object descriptions in the WKF. The difficulty centers around how to convert from the spatial, visual, syntactic representation of the SFF to the more general, modality-independent, semantic representation of the WKF.

In order to deal with this technical issue in more detail, it is necessary to introduce a computational formalism in which the Four Frames model can be expressed. The massive amount of continually interacting information within the model argues for a highly distributed computational formalism. The following section presents a *connectionist* formalism [Feldman & Ballard, 1982; Feldman, 1981] which has been shown to be adequate for expressing complex information-processing models of interesting behaviors.

## 2. Connectionist Models

Connectionism provides a powerful formalism for expressing the massively parallel computation required by many complex information-processing tasks. The full elaboration of the approach can be found in [Feldman & Ballard, 1982] and [Feldman, 1981]. Only those aspects of the model which are relevant to the particular connectionist structures used in our recognition model are presented here.

A connectionist network consists of a large collection of active *units* widely interconnected by *links* (or *connections*). The fundamental premise of connectionism is that, in order to adhere to the time constraints imposed by basic perceptual processing, these units do not communicate large amounts of symbolic information. Rather, they transmit very simple information (e.g. activation levels) and compute by being appropriately *connected* to many other similar units.

### 2.1 Units

A unit is a computational entity intended roughly to capture the information-processing capability of a neuron (see Figure 2.1). The formal specification of a unit is presented in the Appendix. A unit receives input via links which have an assiciated positive- or negative-valued *weight*. Input on a link is scaled by the associated weight and combined with other scaled inputs at a particular *input site* according to a *site function*. A unit may have more than one input site, each carrying out local computations whose results are combined to raise or lower the unit's real-valued *potential* and determine its discrete *state*. These values are used to compute the unit's *output*, which typically non-zero only if the unit's potential is above some specified *threshold* level.

Figure 2.1: A prototypical unit

A very useful unit construction involves the use of site functions which respond only to co-occuring activation on all of the input connections at that site. Such *conjunctive connections* can be used to signify that the different aspects of an enabling condition obtain. The unit becomes active (i.e. has state of *firing* and non-zero output) if the conjunctive connections at any of its input sites are active. In this way, units can be constructed which compute the continuous analog of a logical OR-of-AND, therby responding to any of a finite set of distinct enabling conditions. This type of construction will be used extensively in our connectionist mechanisms.

### 2.2 Stable Coalitions and WTA's

Each unit in a connectionist network continually computes its output value using its current state, potential and input values, and sends this output to all units to which it is connected. Some inputs cause a unit to increase its output value, while others may cause a decrease. The network as a whole computes by stablizing into a global state in which a set of active, mutually reinforcing units, called a *stable coalition*, dominates the activity of all other competing units for some time. Such a set of units represents a fixed interpretation of a visual input.

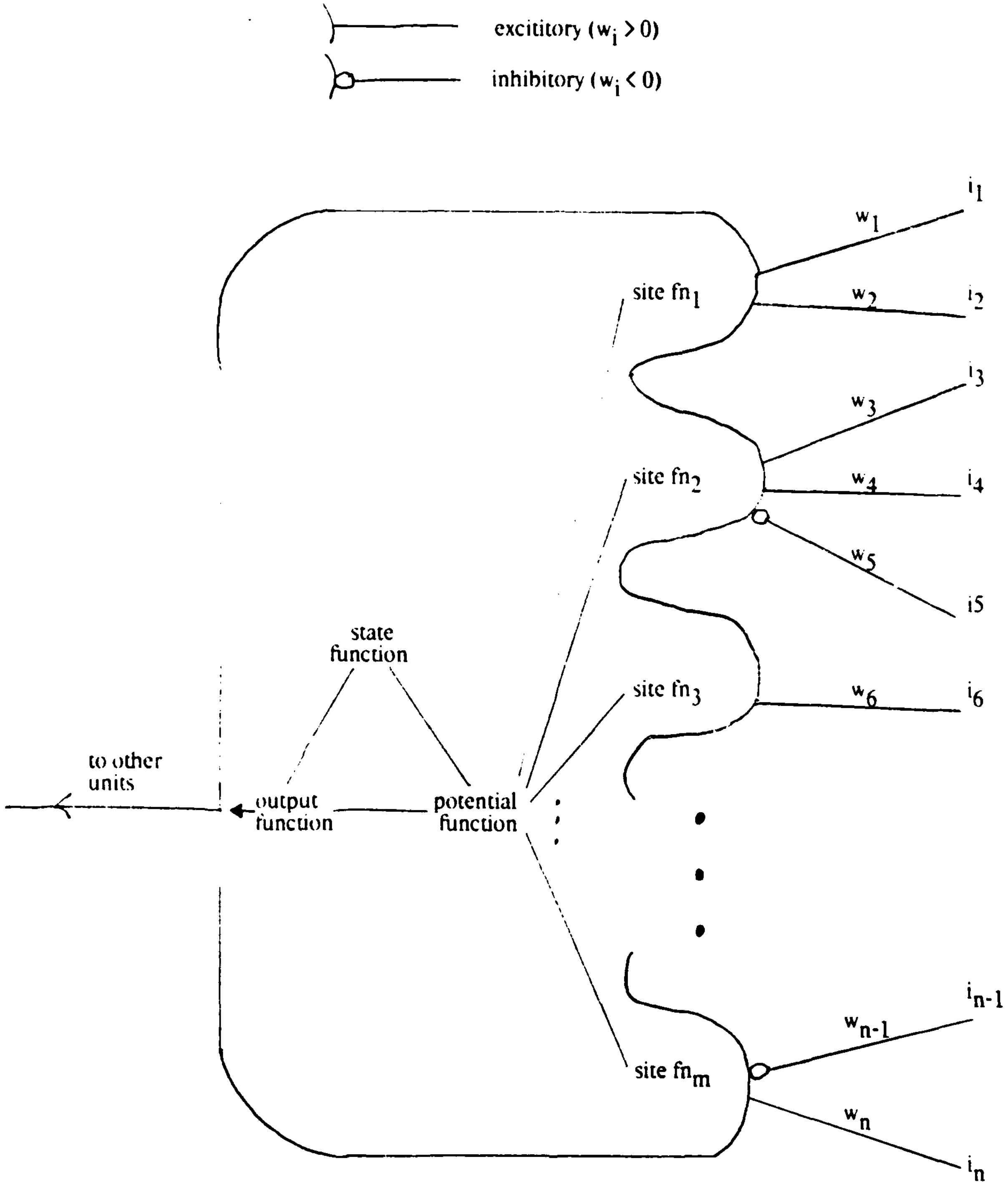


Figure 2.1: Prototypical unit

The significance of allowing negative weights on links becomes clear when we consider the poblem of getting a distributed network to make a decision or choose from a number of alternatives. Units representing alternatives can compete, via interconnecting inhibitory links, until a single unit is active and supressing the activity of all others (see Figure 2.2). Such a construction, known as a *Winner-Take-all network* (WTA), is useful in ensuring the stability of a network representing mutually inconsistent alternatives, only one of which can be active at any time.

Figure 2.2: Winner-Take-All network

## 2.3 Routines

A difficult problem for many models of distributed computation is how to capture within a highly parallel formalism the inherent sequential nature of many types of information processing. The connectionist mechanism for controlling sequential processing is called a *routine* and consists of a set of units connected in series (see Figure 2.3). *Question units* send projections to some knowledge structure and to an *enable unit* which issues an *enable signal* to one or more assiciated *answer units*, which are interconnected in a WTA. (An enable unit is used to ensure that the enable signal to the answer units is constant without requiring that the signal from the question unit to the knowledge structure be constant.) Answer units receive input from the knowledge structure and compete to reach threshold. The answer unit to reach threshold first suppresses the other contenders (and the associated enable unit), and sends activation to the next question unit in the routine. In this way, sequential decisions and actions can be generated which are based on the massively parallel computations in the knowledge structure. Shastri and Feldman [Shastri & Feldman, 1984] develop further the notion of routines. The specific constructions required by our recognition model will be specified in detail in Section 5.

Figure 2.3: Connectionist routine

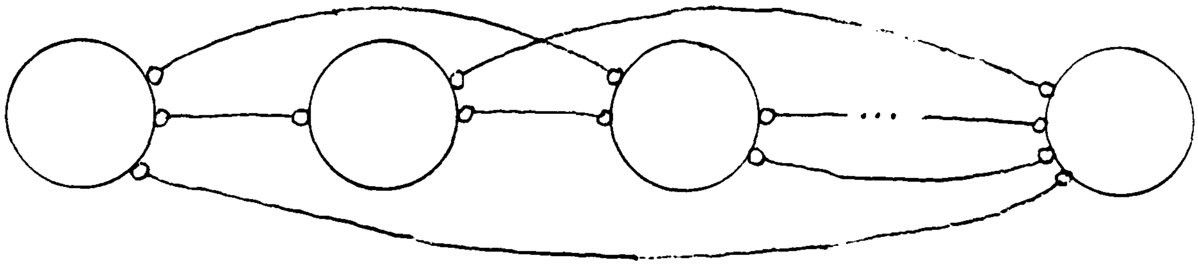


Figure 2.2: Portion of a simple Winner-Take-All network

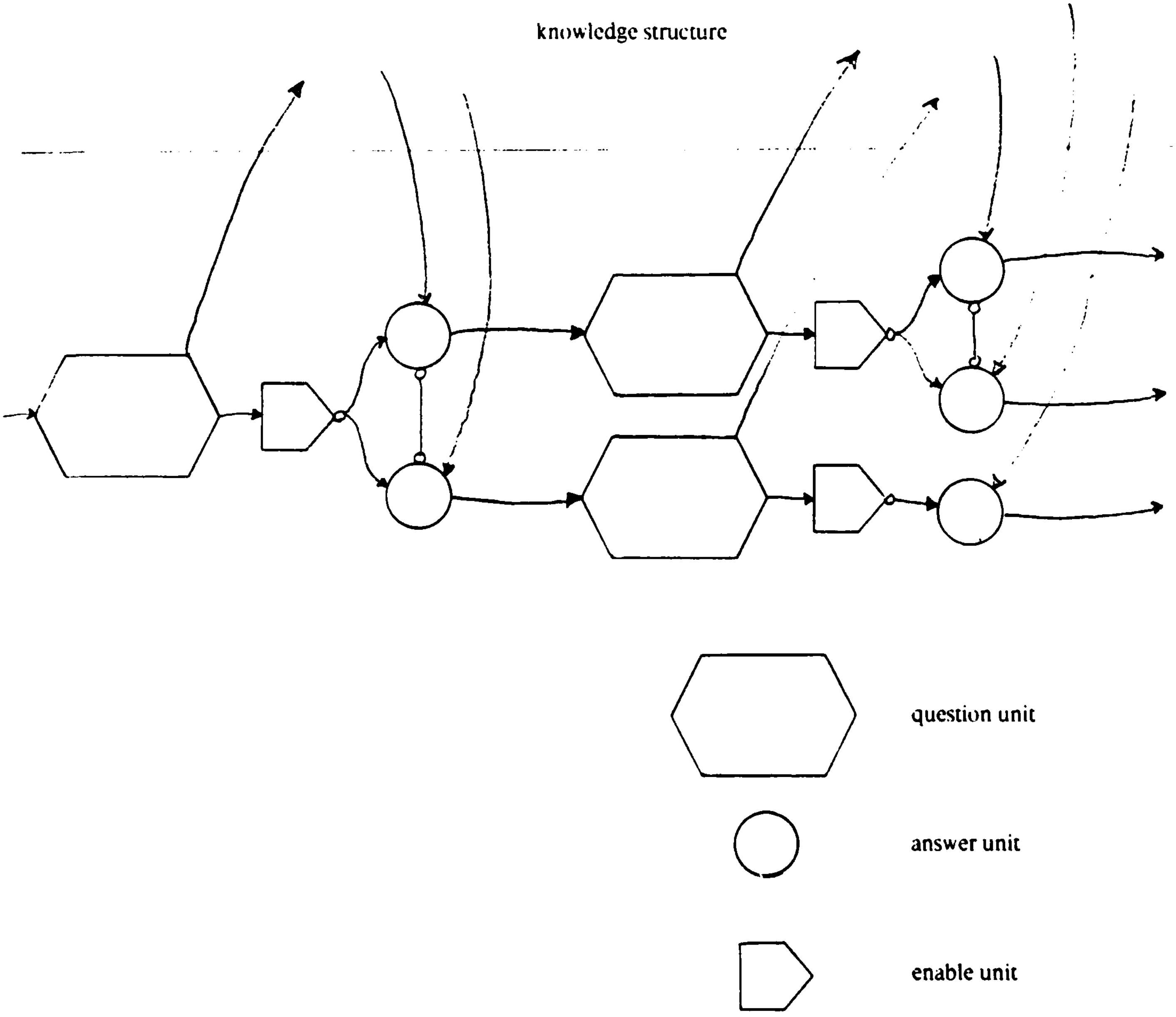


Figure 2.3: Connectionist routine

## 3. The Matching Problem

We are now in a position to examine in more detail the nature of the SFF representation and how feature information might be matched with object representations in the WKF. In order to keep the problems down to a manageable size as well as enable specific examples, a simplified visual domain will be developed which exhibits most of the difficulties of real-world vision that we will want our model to address.

### 3.1 Feature Sets and Visual Elements

Recall that the SFF continually maintains viewer-centered stable feature information. The organizing principle behind the representation of information in the SFF is that all significant visual information is expressed in terms of discrete parameter values in some multidimensional feature space. For our purposes we can assume that each possible value for a feature is explicitly represented for each position in the visual field by a particular unit. Our simplified domain will have *feature-value units* for color (red and brown), texture (dull and shiny), size (large and small), and general shape (spherical and cylindrical) at each position of a hexagonally-arranged 10 by 10 visual field. Features such as size and shape which cover several units are assumed to be represented by a single unit at the center of the region covered. This overly simple version of the *unit-value principle* [Barlow, 1972] will be used in the work to follow, but requires too many units to be plausible at a realistic scale. [Feldman & Ballard, 1982] and [Hinton, 1980] have shown how techniques such as coarse and coarse-fine encoding can reduce the number of units required to within acceptable limits.

Our use of a 10 by 10 visual field in our description of the model is intended to be consistent with the development of the Small World of [Feldman, 1984] and provides us with enough freedom to describe clearly the various spatial interactions in our model. However, the implementation results in Section 5 are based on a visual field with only ten elements, as shown in Figure 5.2.

While the particular choice of feature set used in the SFF is not critical, the relation between these features and object representations in the WKF *is* critical. Specifically, the model assumes that any primitive object can be uniquely characterized by some particular set of values of these features. For this to be plausible, the RF-SFF computations must be sufficient to process light intensity information into a set of features expressive enough to fully describe primitive objects in the domain. Previous work by Ballard [Ballard, 1984] on parameter networks has demonstrated the plausibility of extracting features such as general shape and size, as well as color and texture, from image data without the use of object-specific information. We will assume such parameter networks continually maintain in the SFF the best value for these features at each position in the visual field. This stable feature information will serve as input to our recognition mechanism.

Since each primitive object must be characterizable by a set of feature values, a given feature set inductively defines the set of recognizable primitive objects. We call such objects *visual elements* and represent them as units connected to the appropriate feature-value units in the SFF. For example, the unit for a large, red, shiny cylinder would receive conjunctive connections from the four units representing these feature values at each spatial position (see Figure 3.1). Object representations in the WKF are hierarchically built from these visual elements and will be more fully described in the next section. If the set of feature-value units active at each position activates the corresponding visual element, a complex object present in the image will have most of its primitive subparts active. An object is a candidate for recognition when its representation in the WKF becomes active and forms a stable coalition.

Figure 3.1: Feature-value units connected to visual elements

## 3.2 Crosstalk

A major problem with the simple mechanism described above is common in parallel models of visual processing and is known as *crosstalk* among features. Crosstalk arises from confusion among features at different spatial positions. Thus the simultaneous presentation of a large cylinder and a small sphere might inappropriately activate the large sphere unit. The difficulty arises from the spatial independence of the visual elements. Since the appropriate feature-value units at any position can activate a visual element, the process is insensitive to the particular positions of the features activating a visual element. The naive solution of having a separate unit at each position in the visual field for each visual element clearly requires too many units to be plausible for realistic feature sets.

To retain some spatial coherence of features during the activation of visual elements, [Feldman, 1984] proposes that units representing *pairs* of feature values be used conjunctively to activate these units. These *feature-pair units* have an input site for each position in the visual field and become active if the proper pair of features co-occur at the same spatial position (see Figure 3.2). Feature-pair units are also spatially independent (i.e. there is only one red cylinder unit), but conjunctive activation using a large subset of possible feature pairings enforces a rather strong spatial constraint on feature patterns that will activate a given visual element. This spatial constraint is sufficient to eliminate excessive crosstalk of features during the identification of visual elements.

Figure 3.2: Activating visual elements using feature-pair units

Unfortunately, proper recognition of a complex object (i.e. one composed of subparts, each of which is either another complex object or a visual element) requires higher-order spatial constraints. In particular, the spatial relations between subparts (i.e. the object's internal geometry) must be verified during the recognition process. Otherwise, the scattered pieces of a complex object would be no differently recognized than the actual coherent object. For example, the model would be unable to distinguish between the inputs shown in Figure 4.4. In order to address this issue, we elaborate on the representation of complex objects in the WKF in the next section.

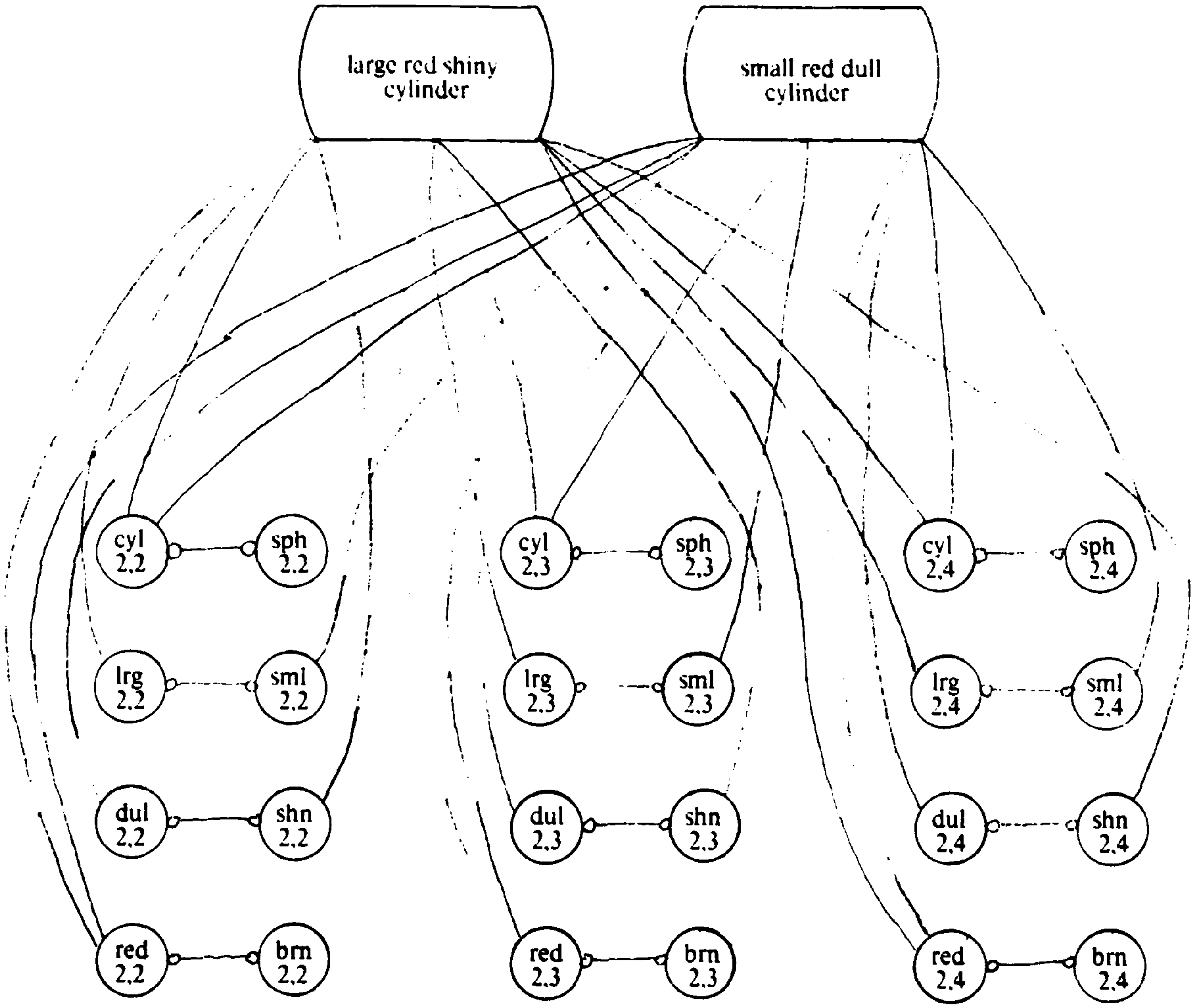


Figure 3.1: Feature-value units connected to visual elements

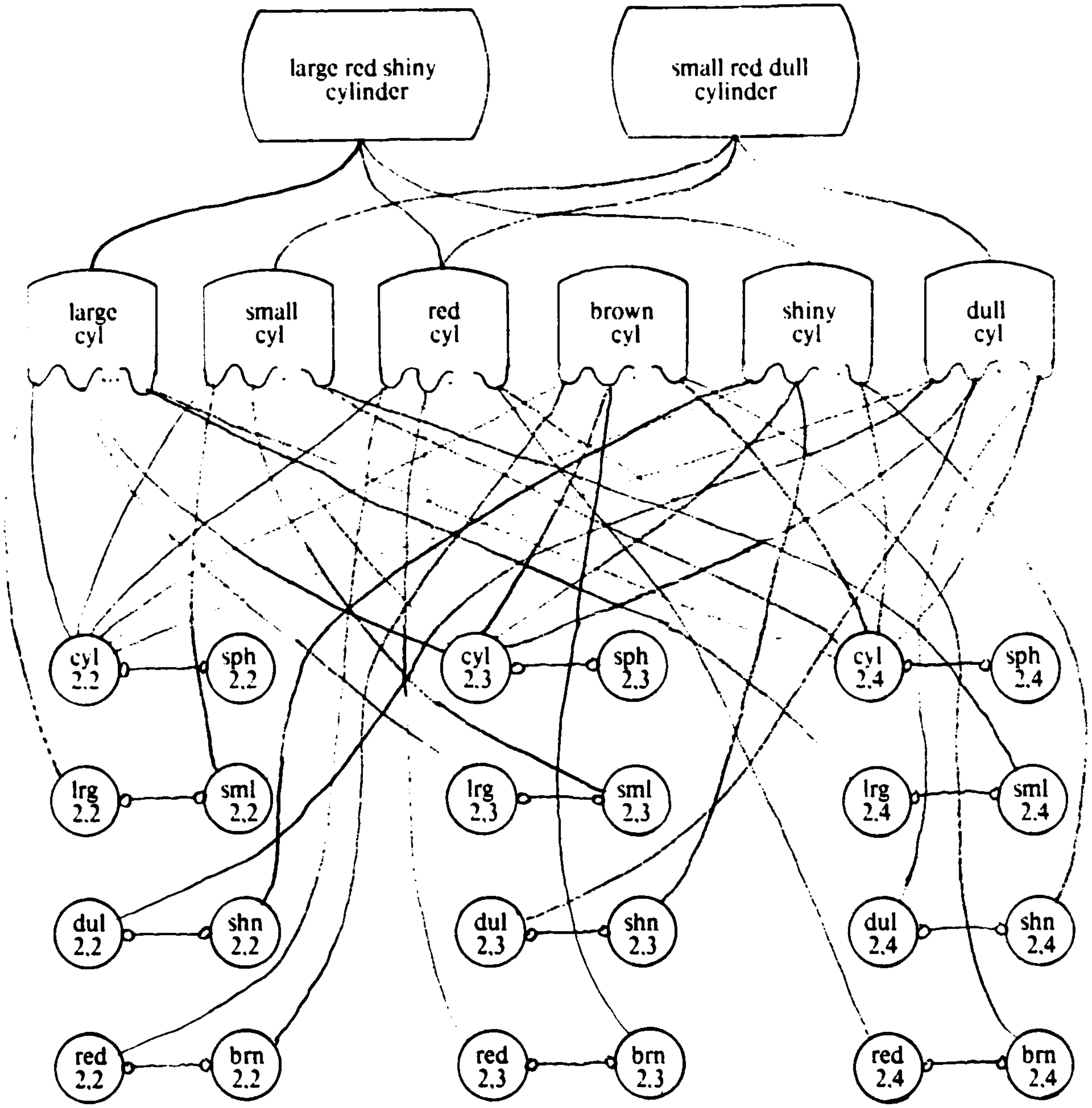


Figure 3.2: Activation of visual elements using feature-pair units

# 4. Verifying Internal Geometry

## 4.1 Object Representations

We mentioned earlier that object representations in the WKF will be hierarchically built from visual elements which can be directly activated by feature input. The objects in our simplified visual domain will have a two-level hierarchy and will consist of simple "toys." For example, a train will consist of a large, red, shiny cylinder with two small, brown, dull spheres below, and a small, red, shiny cylinder on top (see Figure 4.1). Hammers, bears, dumbbells and other toys are similarly characterized. It would be straightforward to extend this domain to higher-leveled hierarchies (e.g. the bear's head might have two eyes, a nose and a mouth), but we prefer to focus on processing details within a single level of the hierarchy. Hrechanyk and Ballard [Hrechanyk & Ballard, 1982] present a connectionist model of form perception which describes the processing details of switching between levels in a hierarchical representation. The current model draws heavily on their work and can be viewed as an elaboration of their model to within-level processing.

Figure 4.1: Some recognizable objects

Thus, at the highest level in an object's WKF representation, an *object unit* will receive input from structures representing the most global components of the object. In our domain, these components will simply be visual elements. Activation of an object unit signifies that all of the subparts of that object have been identified in the image.

In order for the internal geometry of an object to be verified during recognition, the spatial relations between subparts must be represented at each level in the hierarchy. We encode a relation in terms of the distance and direction between two subparts with respect to a canonical (object-centered) reference frame specific to that level of the hierarchy. The reference frames for the various levels are related hierarchically, with each frame specifying the scale, orientation and position of the subparts at that level with respect to the next higher (i.e. more global) reference frame [c.f. Marr & Nishihara, 1978; Hrechanyk & Ballard, 1982]. Each reference frame is chosen to maximize the symmetry and elongation of the description of the subparts at that level [Palmer, 1983]. In our model, we assume that a particular "main" subpart determines the characteristics of the reference for that level, and all subpart relations are encoded with respect to this main subpart. The idea behind this representation is illustrated in Figure 4.2. The actual encoding of these relations in connectionist networks will be markedly different and elaborated on in Section 5.

Figure 4.2: Representation of internal geometry

## 4.2 Viewing Transform

Recal that features in the SFF are represented relative to a *viewer frame* of reference (determined roughly by head position). Given the representation of object components with respect to a canonical *object frame* of reference, the process of mapping image features to object representations requires a *viewing transform* to

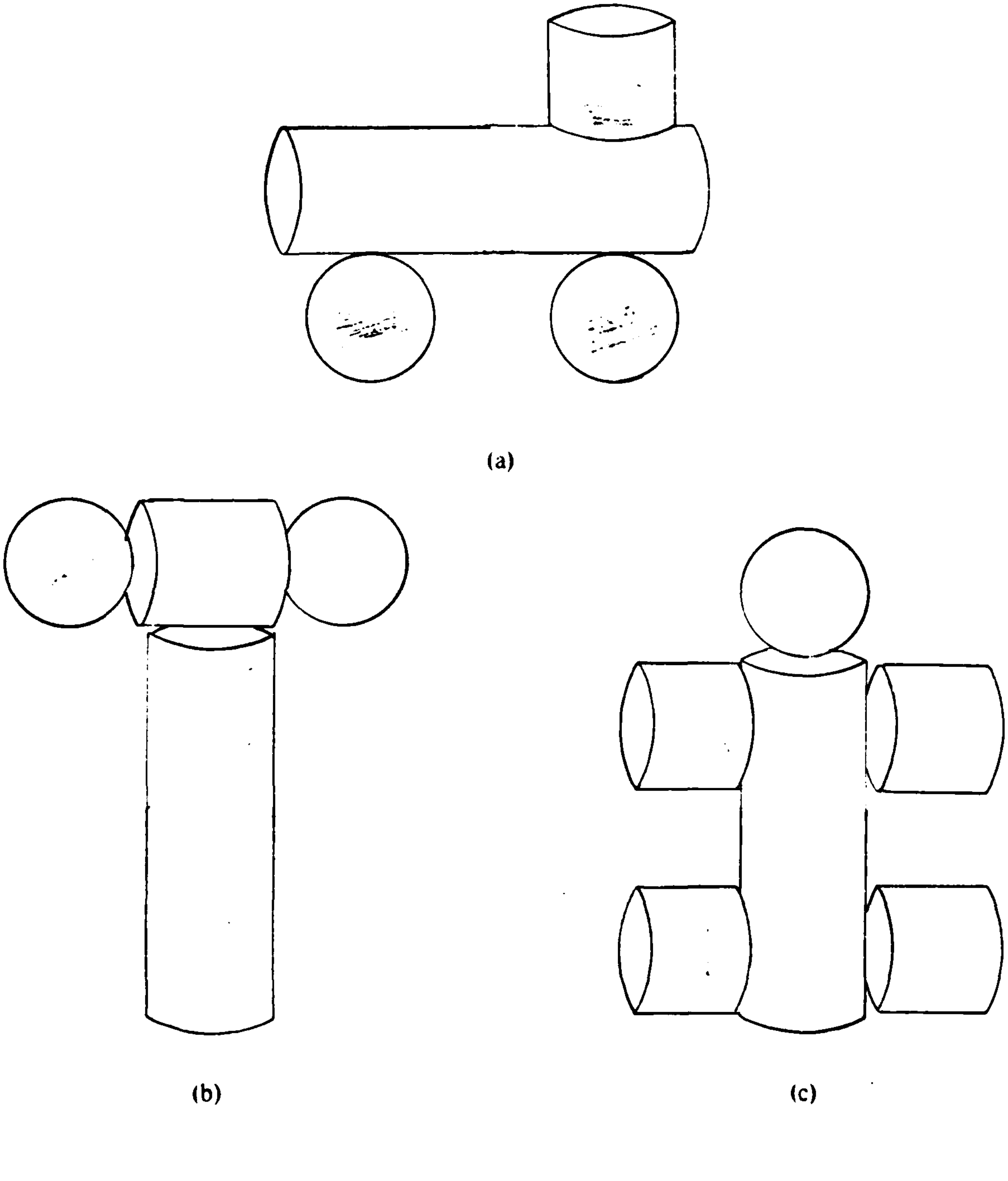


Figure 4.1: Some recognizable objects: (a) a train, (b) a hammer, and (c) a bear.

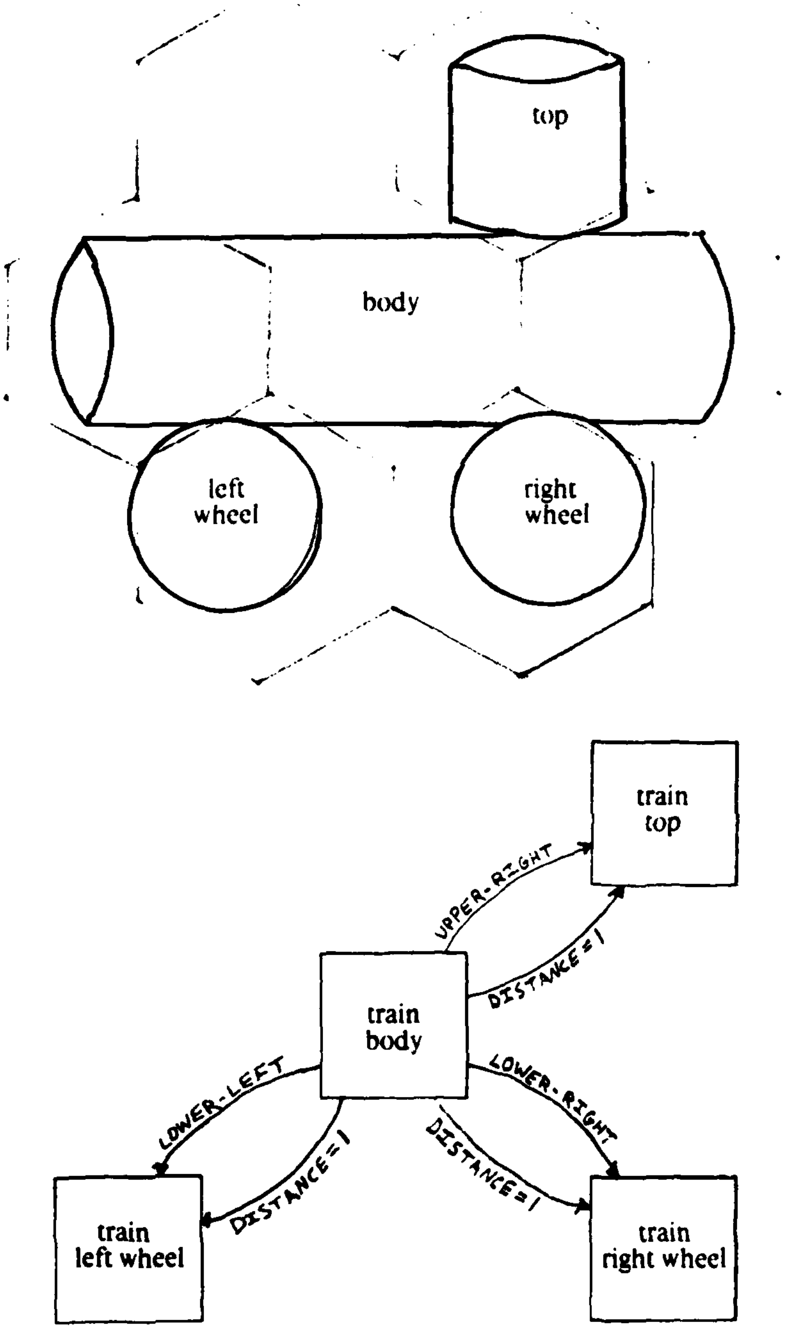


Figure 4.2: Representation of the internal geometry of a train.

relate these two reference frames [Marr & Nishihara, 1978; Hinton, 1981; Ballard & Sabbah, 1982; Hrechanyk & Ballard, 1982]. Specifically, the parameters of dilation, rotation and translation which relate the scale, orientation and position of the object frame with those of the viewer frame must be calculated.

A viewing transform is needed during the verification of subpart relations because the distance and direction between two subparts in the internal representation will be related by this transform to their corresponding features in the image. Figure 4.3 illustrates the relation between the representation and image of an object being viewed in a particular scale, orientation and position.

Figure 4.3: Relation between the representation and image of an object

We can now describe in general terms how the relations between subparts within a level in the hierarchy might be verified. First, the scale, orientation and position of the main subpart in the *image* (i.e. with respect to the viewer frame) must be determed (spatial information was lost during the initial activation of visual elements). Since these parameters are explicit in the internal representation of the main subpart (they define the object frame), the viewing transform parameters of dilation, rotation and translation can be computed. These parameters, in conjunction with the distance and direction information in the representation, define a mapping from the position of the main subpart in the image to the positions of the various other subparts at that level in the hierarchy. Using this mapping enables us to verify that the subparts are in the appropriate positions in the image, and hence, in the proper spatial relations with one another.

It should be noted at this point that, although the design of the model allows any number of possible values for the transform parameters, the scale of the implementation described in Section 5 severely limits the range of these parameters. In particular, only a single value of dilation is supported in the implementation.

### 4.3 Relational Crosstalk and Sequentiality

A difficulty with the above approach is that there is no way to ensure that the relations hold between the proper pair of subparts. Thus the two images in Figure 4.4 would both be identified as a train. This crosstalk of relations arises because the system described above attempts to verify the distance and direction relations for the various subparts *simultaneously*. Resource constraints on the number of units and connections available to encode the viewing transform rule out a separate mechanism for each relation. Furthermore, psychological evidence for interference of transformations during object perception [P. Jolicoeur, in preparation] suggests, in fact, that observers have only one such mechanism.

Figure 4.4: Two images identified as a train

To avoid this crosstalk, the subpart relations at each level in the hierarchy must be verified *sequentially*. After the viewing transform parameters are determined, the relational information for a *single* subpart will be used to determine the position in the image where that subpart should be found. The features in the image at that position can then be used to determine the visual element present at that position.

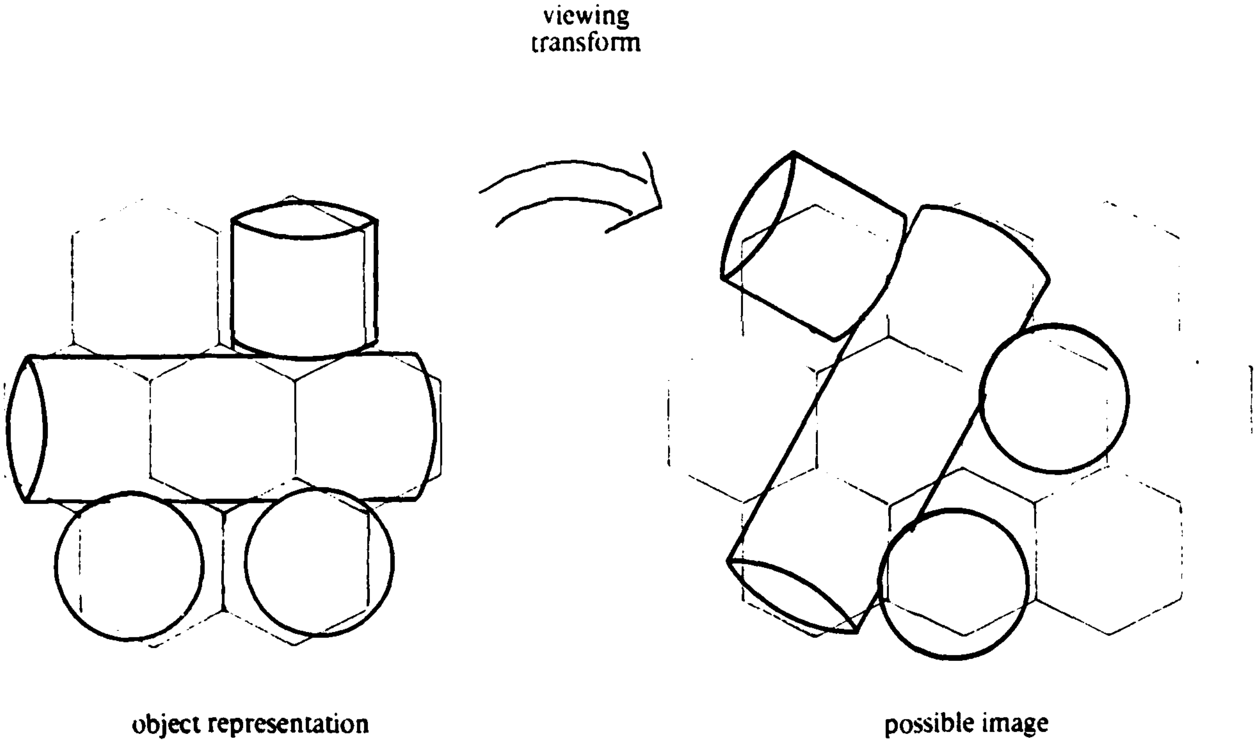


Figure 4.3: Viewing transform relating object representation to image features

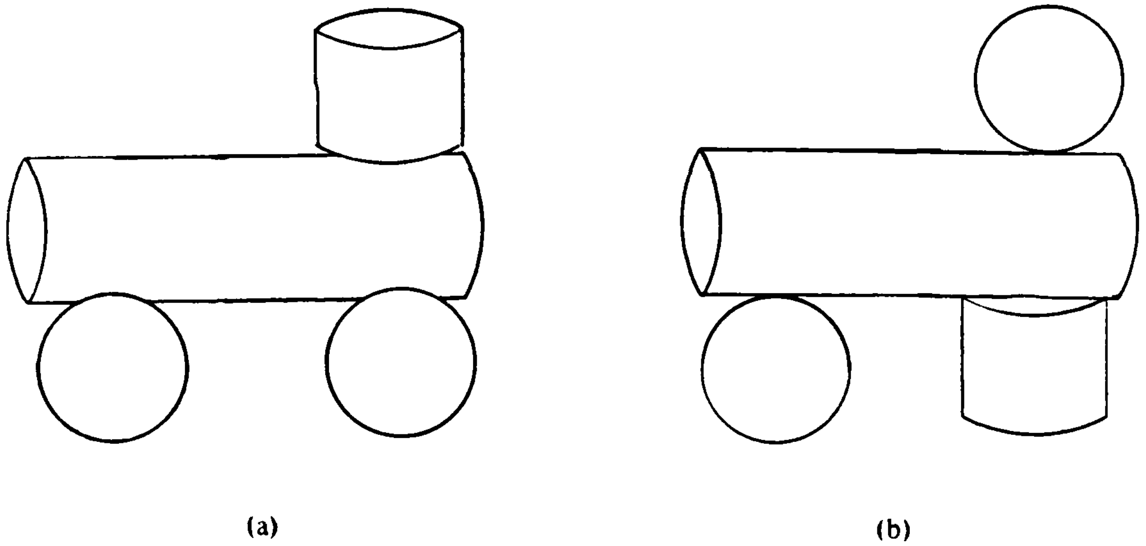


Figure 4.4: Two images that would be identified as a train if subpart relations were verified simultaneously.

The relation is verified if the appropriate visual element for the subpart is activated. Each subpart relation for that level of the hierarchy is verified similarly in turn.

# 5. A Connectionist Model of Object Recognition

All of the necessary groundwork has been laid for developing a connectionist model of how SFF feature information is associated with object representations in the WKF. While the model will be specified in the context of the simplified visual domain we have developed, the mechanisms involved were designed to adhere to the constraints of real-world vision. We consider the overall operation of the model to be useful in understanding how visual feature information might be used to access knowledge of the visual appearance of objects.

## 5.1 Overview

The model consists of three interacting computational subsystems: (1) a *subpart identification mechanism* (SIM) within the SFF for verifying that the visual elements of an object are present in the image; (2) a *viewing transform mechanism* (VTM) for computing the transform which maps an object frame of the WKF to the viewer frame of the SFF so subpart relations can be verified; and (3) a collection of *object representations* in the WKF, each consisting of a hierarchy of visual elements and a routine connected with both the SIM and VTM controling the sequential verification process.

The organization of the model is shown schematically in Figure 5.1. The SIM determines which visual elements are present in the image and activates the routine of an object whose subparts are successfully identified (1). Since spatial information is lost during this process, the routine must determine the scale, orientation and position of the main subpart of the object using top-down feedback to the SIM (2). This information is combined with the relational information of the object frame to determine the dilation, rotation and translation parameters of the viewing transform (3). The routine then sequentially activates the relational information for a particular next subpart (4), which the VTM uses to map to the position in the image which should contain that subpart (5). The SIM selectively activates a visual element using feature information at this position only, and signals to the routine if the appropriate subpart is activated (6). The routine attempts to verify each subpart relation in turn (return to (4)). If all relations are verified, a final unit is activated signaling to the rest of the visual system that the object has been recognized (7).

Figure 5.1: General organization of the recognition model

In presenting the detailed operation of the connectionist model, we omit many of the computational details (e.g. internal operation of units), prefering to include them in the Appendix. In addition, we will make the simplifying assumption that only a single object is present in the image at one time. Our model assumes that additional *focus-of-attention* mechanisms (as described in [Feldman, 1984]) restrict processing to the portion of the SFF receiving input (via the *gaze* mapping) from the foveal region of the RF. This restriction prevents excessive interference within the VTM caused by different object routines simultaneously attempting to verify subpart relations. Furthermore, we will not explicitly address the issues involving the processing between levels in a multi-leveled hierarchical object representation. Hrechanyk and Ballard [Hrechanyk & Ballard, 1982] have argued that the different levels of a

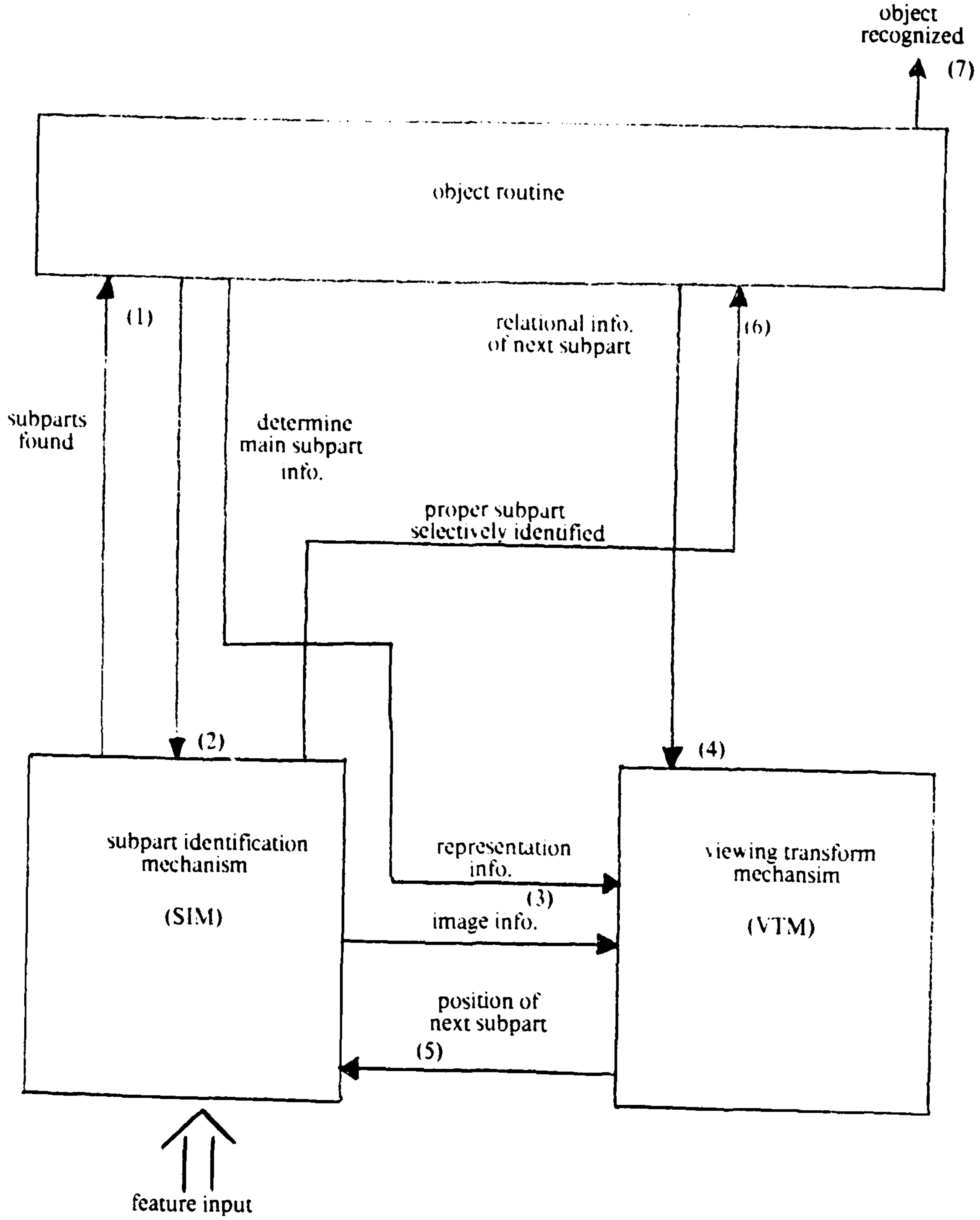

object recognized
(7)
object routine
(1)
subparts found
determine main subpart info.
relational info. of next subpart
(6)
proper subpart selectively idenufied
(2)
(4)
subpart identification mechanism
(SIM)
representation info.
(3)
image info.
viewing transform mechansim
(VTM)
position of next subpart
(5)
feature input

Figure 5.1: General organization of recognition model

hierarchy must be verified sequentially, and have specified a connectionist model for relating the information at each successive level in a way compatible with the single-level processing of our model.

### 5.2 Subpart Identification

Figure 5.2 illustrates a possible section of visual input and the portion of the SIM for processing this input. Lower-level processes (parameter networks) activate the feature-value units corresponding to feature information in the image. Pairs of feature-value units at the same position project to a particular input site of the corresponding feature-pair unit. A feature-pair unit becomes active if any of its input sites receives activation from both of its associated feature-value units (i.e. those features co-occur anywhere in the image). These feature-pair units, in turn, project to the appropriate visual elements, which become active if they receive input from each of their component feature-pair units. Since our object representations have only a two-leveled hierarchy, visual elements project directly to object units representing objects to which they belong. If all of the subparts of an object are present in the image, its object unit will become active and send an enable signal to the routine for verifying the internal geometry of the proposed object (see Figure 5.2).

Figure 5.2: Organization of the SIM

An important aspect of the structure of the SIM is that all connections between units are *bidirectional*: any unit connected to another unit also receives a similar connection from that unit. These reciprocal connections allow mutually consistent peices of information to reinforce each other. In addition, connections from higher to lower levels enable the top-down feedback required to recover the spatial information lost during the identification of visual elements.

### 5.3 Determining Viewing Transform Parameters

The parameters of dilation, rotation and translation for the viewing transform are computed by relating the scale, orientation and position of the main subpart in the image with its canonical description in the internal representation. The different values for each of the transform parameters will be represented by units interconnected in a WTA, so only one value for each is active at any time. The computations required to determine these parameters are represented in Figure 5.3.

Figure 5.3: Computations of viewing transform parameters

To recover the needed spatial information, the first unit in the routine for the object sends top-down activation to its object unit in the WKF, which, in turn, activates the visual element corrresponding to the object's main subpart. This visual element unit projects down to activate its associated feature-pair units.

Since the feature-pair units are spatially independent, they send activation to the two corresponding feature-value units at every position in the SFF. Not all such units become active, though. Only those feature-value units that were activated by visual input *and* are now receiving top-down input become active. (Information

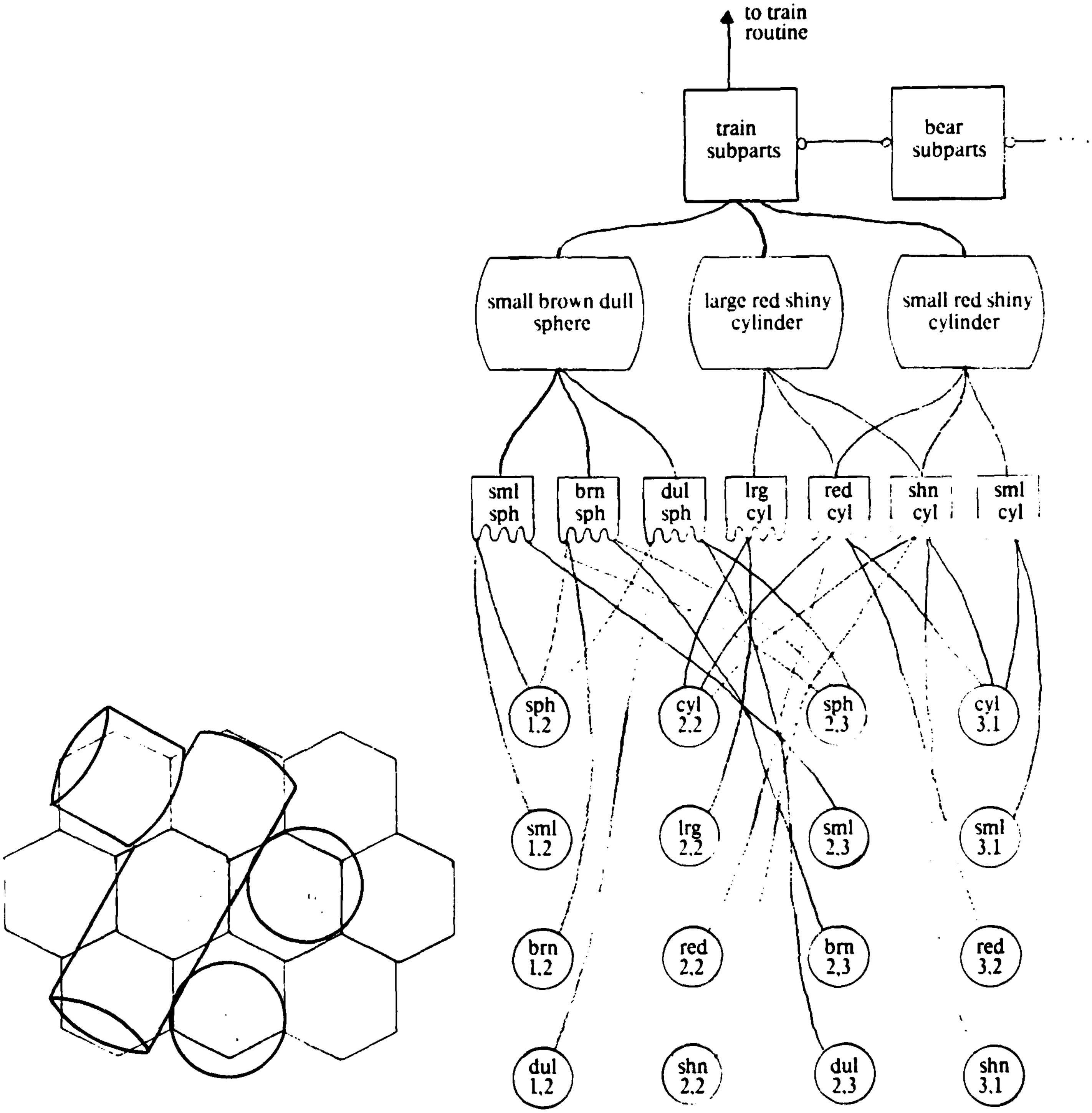


Figure 5.2: (a) possible feature input, (b) organization of portion of SIM for analyzing such input.

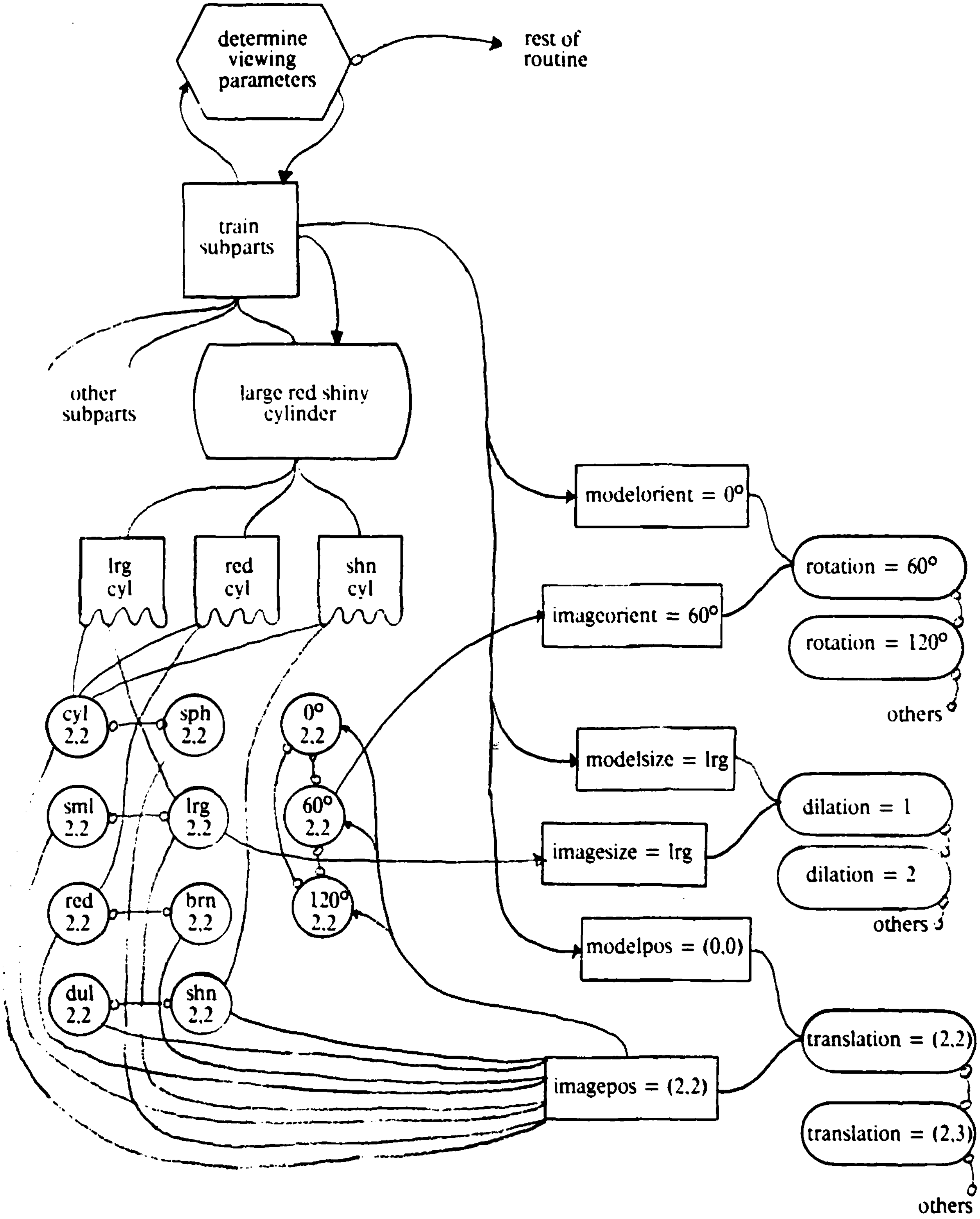


Figure 5.3: Determination of viewing transform parameters

regarding whether a feature-value unit was activated by visual input is stored in its *state*, as explained in the Appendix.) This intersection of visual input and appropriate feature information activates every feature value unit representing a feature that is among those of the main subpart (top-down constraint) and is present in the image at the unit's position (bottom-up constraint). Thus a position with feature-value units active for each type of feature contains all of the feature values of the main subpart, and hence represents the position in the image in which the main subpart appeared. This position is computed by *image parameter units* whose sites receive disjunctive input from each pair of feature-value units at a position. The unit becomes active if each of its sites receives activation (see Figure 5.3).

In order to determine the translation needed in the viewing transform, we must compare this position with the position of the main subpart in the internal representation. Since the characteristics of the object frame are relative to the main subpart, we assume that the frame is centered on this subpart (i.e. the main subpart has position (0,0) in the internal representation). Thus the needed translation is fully determined by the position of the main subpart in the image (which we just calculated). The object unit activates a *model parameter unit* representing the canonical position of the object (i.e. (0,0)). Activation from this unit enables the image parameter unit representing the position of the main subpart in the image to activate the appropriate translation unit. For example, if the main subpart is in the position (2,2), the translation unit for (2,2) would be activated.

The rotation required by the viewing transform is calculated in a similar manner. Image orientation information is assumed to be available as feature values in the SFF as a result of parameter network calculations for shape [Ballard & Sabbah, 1982]. Top-down activation of the main subpart will not activate the unit representing its orientation, however, since orientation is not a feature used in the identification of visual elements. Therefore, this information must be recovered *after* the position of the main subpart is determined. The image parameter unit representing the main subpart's position sends activation to the orientation feature-value units at that position. As with other feature-value units, only the unit which was activated by visual input now becomes active. This unit, in turn, activates an image parameter unit representing the orientation of the main subpart in the image.

Whereas the position of the main subpart of every object is assumed to be (0,0), the main subpart may be at one of a number of different orientations in the WKF representation of the object (i.e. the main subpart of a train is horizontal, while for a hammer it is vertical). A model parameter unit representing this canonical orientation is activated by the object unit. Activation from this unit conjoins with the activation from the image parameter unit for orientation to activate the appropriate rotation unit. For example, an orientation of 0° in the representation and 60° in the image would activate the rotation unit for 60°. Notice that the direction of the mapping is from the internal model to the image.

Finally, the dilation parameter of the viewing transform is computed by comparing the feature value for size activated by the top-down activation of the main subpart with size information in the internal representation. The active size feature-value unit activates an image parameter unit representing the size of the main subpart in the image, while the object unit activates a model parameter unit representing its canonical size. Activation from these two units conjoins to activate the appropriate dilation unit (see Figure 5.3). Thus, a size of *small* in the model and *large* in the image would imply a dilation of 2.

Once the units representing the parameters of the viewing transform become active, they remain active throughout the sequential verification of subpart relations. They also strongly inhibit competing units to eliminate interference within the VTM.

## 5.4 Verifying Subpart Relations

The structure of the VTM is illustrated in Figure 5.4. The purpose of this mechanism is to determine the position in which a subsequent subpart should appear in the image given its relation to the main subpart and the calculated viewing transform parameters. Units representing distance and direction relations project, along with transform parameter units, to *binder units* which, in turn, project to *position units.* Binder units become active if they receive input on all three incoming connections. These units encode two sets of constraints: (1) *distance* constraints which, given a translation, distance and dilation, determine a set of positions; and (2) *direction* constraints which, given a translation, direction and rotation, detemine a second set of positions. For example, a translation of (2,2), distance of 2, and dilation of 2, specifies the circle of positions at a distance of 4 from the position (2,2). Likewise, a translation of (2,2), direction of upper-right, and rotation of -60° specifies the positions to the right of (2,2). The intersection of these sets determines the unique position (in this case, (2,6)) which adheres to both constraints (i.e. the position at the appropriate distance and direction given the viewing transform). To carry out this intersection, each position unit receives input from pairs of binder units (one for each type of constraint) and becomes active if both units in some pair becomes active and send input.

Figure 5.4: Viewing transform mechanism

To verify a subpart relation, a question unit representing the relation sends activation to the distance and direction units representing the subpart's relation to the main subpart in the canonical representation. The question unit also activates an enable unit which sends an enable signal to a single associated answer unit. The distance and direction units send activation to the binder units to which they are connected, some of which are receiving input from the active tranform parameter units. Certain binder units thereby become active and send activation to position units. A single position unit will receive input from both members of a binder unit pair and become active. This unit represents the position that the subpart should be in given the viewing transform.

In order to verify that the subpart is actually in this position in the image, the position unit sends activation to all feature-value units in the SIM at its position (see Figure 5.4). These units become active only if they were activated directly by visual input *and* are now receiving activation from the VTM (using the same internal operation as before when determing the image parameters of the main subpart). The active feature-value units at this position activate consistent feature-pair units which, in turn, attempt to activate a visual element. The visual element appropriate for the subpart is connected to the answer unit attempting to verify its relation to the main

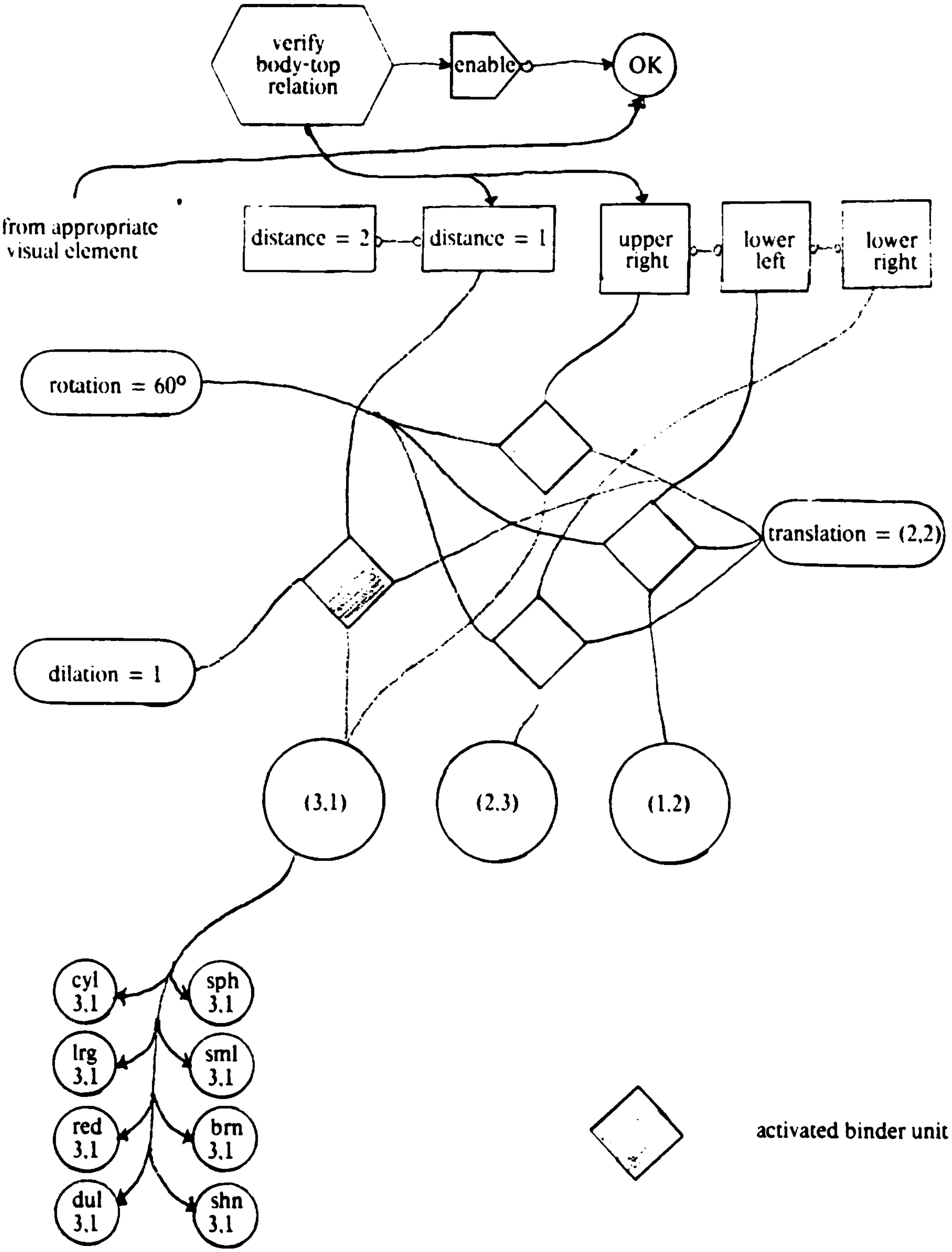


Figure 5.4: Structure of the VTM

subpart. Since only one position in the SIM is receiving input from the VTM, if the active feature-values succeed in activating this visual element, then this subpart must be in the proper position in the image. and, hence, in the proper relation to the main subpart. In this case, the answer unit in the routine will become active, signaling that the relation has been verified.

The activated answer unit inhibits its associated enable unit, and activates the next question unit in the routine. This question unit attempts to verify the relation of the next subpart to the main subpart. The answer unit for the last relation in the routine activates a final unit which signifies that all relations have been verified and the object has been successfully recognized.

### 5.5 Implementation Results

The recognition model that we have described was implemented using the ISCON connectionist simulator [Small *et at.*, 1983]. Because of ISCON's prohibitive slowness in simulating large networks, the model was implemented only in as much detail as required to verify that it is computationally sound. Thus, a visual field with only 10 hexagonally-arranged positions is used (see Figure 5.2). This allows the image of an object to be in one of two positions ((2,2) or (2,3)), three orientations (0°, 60°, or 120°) but only one scale (1). This in turn implies that the VTM requires two values for translation ((2,2) and (2,3), three values for rotation (0°, 60° and -60°), and a single value of dilation (1). Furthermore, the internal geometry of an object is represented using three possible distances (1, 2, and 3 units), and six possible directions (left, upper-left, upper-right, right, lower-right, and lower-left). Intermediate directions are represented by weighted combinations of these six. A description of the potential, state, and output fuctions, decay and threshold constants, and connections weights used in the implementation can be found in the Appendix.

The results presented here are actually from an implementation of an earlier version of the model which had visual elements directly connected to the first element of the routine of an object (without object units). Subsequent simulations of the updated version of the model have verified that its dynamics and operation are essentially equivalent to those of the model implemented here.

Tables 5.1 through 5.5 present the results of a simulation of our recognition model using the feature input of a train shown in Figure 5.2. In addition to the structures needed to recognize a train, the structures for a bear and a dumbbell are also included in the implementation. However, as the results show, the routines for these objects never become active. Due to space limitations, not all of the units at a particular stage in the processing are shown (the implementation involves over 200 units and 1100 connections), but the activation levels of all relevant units are shown. In these tables, $t$ is the current output cycle, $p$ and $f$ represent the states of *primed* and *firing*, respectively, and the activation and state of all units with non-zero output is printed in bold.

Table 5.1 shows the initial subpart identification process and subsequent recovery of spatial information about the main subpart of the object present in the image (in this case, the body of a train). At t = 1, the feature-value units receiving input from lower-level processes of the image are active and firing. All other units are inactive.

Feature-pair units are then activated (t=2), which, in turn, activate the visual elements of the object (t=3). Notice that after firing, the feature-value units settle into a *primed* state. Since the three visual elements of a train have been identified in the image, the first unit in the train routine becomes active at t=4. Since neither the bear nor the dumbbell has all of its subparts present in the image, the routines for these objects remain inactive.

Table 5.1: Subpart identification

Top-down activation from this first unit at t=5 activates the visual element representing the main subpart of a train (i.e. the large, red, shiny cylinder unit). This unit, in turn, activates the set of consistent feature-pair units (t=6). At t=7, the feature information of the main subpart is recovered using the intersection of visual input and top-down activation. Notice that in addition to the appropriate feature values at (2,2), consistent feature values at (3,1) are also activated. However, since (2,2) is the only position in which feature-value units are active for *all* the feature types, this position is selected as that of the main subpart of the train. In addition, feature information for size at this position (*lrg(2,2)*) activates a unit representing the size of the main subpart in the image. This occurs at t=8 as shown in Table 5.2.

Table 5.2: Setting viewing transform parameters

Recall that the first unit in the train routine, in addition to activating the visual element for the main subpart, activates the next unit in the routine (which determines the viewing transform parameters). This unit activates the parameters of position, orientation, and size of the main subpart in the internal representation, as shown in Table 5.2. Activation from these model parameter units conjoins at t=9 with activation from the image parameter units for position and size to activate units representing the appropriate values of translation ((2,2)) and dilation (1). Also notice that activation from the image parameter unit for position has activated the feature-value unit representing the orientation of the main subpart in the image (i.e. 60°). This unit activates the appropriate image parameter unit for orientation at t=10. Activation from this unit and the active model parameter unit for orientation conjoins at t=11 to activate the unit representing the value of rotation needed in the viewing transform (60°).

By t=12, units representing the fact that each of the viewing parameters has been set (i.e. *rotationset*, *dilationset*, and *translationset*) have become active (these units receive input from the transform parameter units of the appropriate type). At t=13, these three units conjoin to activate a unit signaling that *all* transform parameters have been set (i.e. *paramsset*), which then activates the answer unit in the train routine for verifying that these parameters have been set. This answer unit activates the question unit attempting to verify the first subpart relation (train body to train top) at t=15, as shown in Table 5.3.

Table 5.3: Verification of body-top relation

In order to verify that the train top is in the proper position in the image, at t=16 the question unit for this relation activates units representing the train top's distance (1) and direction (upper-right) from the train body in the canonical

## Table 5.1: Subpart identification

| | t=1 | t=2 | t=3 | t=4 | t=5 | t=6 | t=7 |
|---|---|---|---|---|---|---|---|
| sml(1,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| brn(1,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| dul(1,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| sph(1,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| lrg(2,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| red(2,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| shn(2,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| cyl(2,2) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| sml(2,3) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| brn(2,3) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| dul(2,3) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| sph(2,3) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| lrg(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 |
| sml(3,1) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | 4.5p |
| red(3,1) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| brn(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dul(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| shn(3,1) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1p | 4.7p | **9.5f** |
| cyl(3,1) | **10.0f** | 7.9p | 6.5p | 5.6p | 5.1 | 4.7p | **9.5f** |
| sph(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| lrgcyl | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | **10.0f** | 6.5 |
| smlcyl | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 |
| redcyl | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | **10.0f** | 6.5 |
| brncyl | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| dulcyl | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| shncyl | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | **10.0f** | 6.5 |
| lrgsph | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| smlsph | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 |
| brnsph | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 |
| redsph | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| dulsph | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 |
| shnsph | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| lrgredshncyl | 0.0 | 0.0 | **10.0f** | 6.5 | **10.0f** | 6.5 | **10.0f** |
| lrgbrndulcyl | 0.0 | 0.0 | 3.3 | 2.1 | 1.4 | 0.9 | 3.7 |
| smlredshncyl | 0.0 | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | **8.4f** |
| smlredshnsph | 0.0 | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 |
| smlbrndulcyl | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| smlbrndulsph | 0.0 | 0.0 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 |
| trainpartsOK | 0.0 | 0.0 | 0.0 | **10.0f** | 0.0 | 2.5 | 0.0 |
| bearpartsOK | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dumbbellpartsOK | 0.0 | 0.0 | 0.0 | 3.3 | 2.1 | 1.4 | 0.9 |

Table 5.2: Setting viewing transform parameters

| | t=7 | t=8 | t=9 | t=10 | t=11 | t=12 | t=13 | t=14 |
|---|---|---|---|---|---|---|---|---|
| lrg(2,2) | **9.5f** | 7.5p | 6.3p | 5.5p | 5.0p | 4.6p | 4.4p | 4.3p |
| sml(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| red(2,2) | **9.5f** | 7.5p | 6.3p | 5.5p | 5.0p | 4.6p | 4.4p | 4.3p |
| brn(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dul(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| shn(2,2) | **9.5f** | 7.5p | 6.3p | 5.5p | 5.0p | 4.6p | 4.4p | 4.3p |
| cyl(2,2) | **9.5f** | 7.5p | 6.3p | 5.5p | 5.0p | 4.6p | 4.4p | 4.3p |
| sph(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0°(2,2) | 0.0 | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| 60°(2,2) | 4.5p | 7.5p | **9.2f** | 7.4p | 6.2p | 5.4p | 4.9p | 4.6p |
| 120°(2,2) | 0.0 | 0.0 | 5.0 | 3.3 | 2.1 | 1.4 | 0.9 | 0.6 |
| imageorient=0° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| imageorient=60° | 1.2 | 0.8 | 0.5 | **9.5f** | 6.2 | 4.0 | 2.6 | 1.7 |
| imageorient=120° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| imagesize=lrg | 1.2 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 | 0.8 |
| imagesize=sml | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | 0.1 |
| imagepos=(2,2) | 1.2 | **10.0f** | 6.5 | 4.2 | 2.7 | 1.8 | 1.2 | 0.8 |
| imagepos=(2,3) | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | 0.1 |
| modelorient=0° | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| modelorient=60° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| modelorient=120° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| modelsize=lrg | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| modelsize=sml | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| modelpos=(0,0) | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| rotation=0° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| rotation=60° | 0.0 | 0.0 | 0.0 | 0.0 | **9.5f** | **9.5f** | **9.5f** | **9.5f** |
| rotation=-60° | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dilation=1/2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| dilation=1 | 0.0 | 0.0 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| dilation=2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| translation=(2,2) | 0.0 | 0.0 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| translation=(2,3) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| rotationset | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **9.5f** | **10.0f** | **10.0f** |
| dilationset | 0.0 | 0.0 | 0.0 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| translationset | 0.0 | 0.0 | 0.0 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| paramsset | 0.0 | 0.0 | 0.0 | 0.0 | 6.7 | 6.7 | **9.8f** | **10.0f** |
| traineparams | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | 0.2 |
| trainparamsOK | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **9.8f** |

Table 5.3: Verification of body-top relation

| | t=15 | t=16 | t=17 | t=18 | t=19 | t=20 | t=21 | t=22 |
|---|---|---|---|---|---|---|---|---|
| traincktop | **9.8f** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| trainetop | 0.0 | **9.8f** | **9.8f** | **9.8f** | **9.8f** | **9.8f** | **9.8f** | 0.8 |
| traintopOK | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **9.0f** |
| distance = 1 | 0.0 | **9.8f** | 6.4 | 4.2 | 2.7 | 1.8 | 1.1 | 0.7 |
| direction = UR | 0.0 | **9.8f** | 6.4 | 4.2 | 2.7 | 1.8 | 1.1 | 0.7 |
| rotation = 60o | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** |
| dilation = 1 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| translation = (2,2) | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| 1from(2,2) | 5.0 | 5.0 | **9.9f** | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| 2from(2,2) | 5.0 | 5.0 | 7.4 | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| Lfrom(2,2) | 4.9 | 4.9 | 4.8 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| ULfrom(2,2) | 4.9 | 4.9 | **9.8f** | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| URfrom(2,2) | 4.9 | 4.9 | 7.4 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| Rfrom(2,2) | 4.9 | 4.9 | 7.4 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LRfrom(2,2) | 4.9 | 4.9 | 4.8 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LLfrom(2,2) | 4.9 | 4.9 | 4.8 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| position(1,1) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1,2) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1,3) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,1) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,3) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,4) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,1) | 0.0 | 0.0 | 0.0 | **9.9f** | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,2) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,3) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| lrg(3,1) | 0.2 | 0.1 | 0.1 | 0.0 | 5.0 | 3.2 | 2.1 | 1.4 |
| sml(3,1) | 4.0p | 4.0p | 4.0p | 4.0p | **8.9f** | 7.2p | 6.1p | 5.4p |
| red(3,1) | 4.2p | 4.1p | 4.1p | 4.0p | **9.0f** | 7.2p | 6.1p | 5.4p |
| brn(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 4.9 | 3.2 | 2.1 | 1.6 |
| dul(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 4.9 | 3.2 | 2.1 | 1.6 |
| shn(3,1) | 4.2p | 4.1p | 4.1p | 4.0p | **9.0f** | 7.2p | 6.1p | 5.4p |
| cyl(3,1) | 4.2p | 4.1p | 4.1p | 4.0p | **9.0f** | 7.2p | 6.1p | 5.4p |
| sph(3,1) | 0.0 | 0.0 | 0.0 | 0.0 | 4.9 | 3.2 | 2.1 | 1.6 |
| lrgcyl | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | 4.5 | 2.9 | 1.9 |
| smlcyl | 0.3 | 0.2 | 0.1 | 0.1 | 0.0 | **9.0f** | 5.9 | 3.8 |
| redcyl | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | **9.0f** | 5.9 | 3.8 |
| brncyl | 0.3 | 0.2 | 0.1 | 0.1 | 0.0 | 4.5 | 2.9 | 1.9 |
| dulcyl | 0.3 | 0.2 | 0.1 | 0.1 | 0.0 | 4.5 | 2.9 | 1.9 |
| shncyl | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | **9.0f** | 5.9 | 3.8 |
| smlredshncyl | 0.8 | 0.5 | 0.3 | 0.2 | 0.1 | 0.1 | **9.0f** | 5.9 |

representation. An enable unit for activating the answer unit for the body-top relation is also activated.

At t=17, this relational information is combined with the viewing transform parameter values to activate the appropriate binder units in the VTM. Specifically, the units *distance=1*, *dilation=1*, and *translation=(2,2)* activate the binder unit *1from(2,2)*, while the units *direction=UR*, *rotation=60°*, and *translation=(2,2)* activate the binder unit *ULfrom(2,2)*. These two binder units uniquely determine the position (3,1), which is where the train top should be in based on the canonical representation of a train and the viewing transform. The unit representing this position is activated at t=18.

At t=19, the position unit for (3,1) activates those feature-value units at (3,1) which were activated by visual input. These units, in turn, activate the consistent feature-pair units (t=20), which then activate a visual element (t=21). Since this visual element is the one appropriate for the subpart being verified (i.e. a train top), at t=22 this unit succeeds in activating the answer unit attempting to verify the body-top relation. At this point (t=23), the answer unit activates the question unit for the next relation to be verified (body-left wheel), as shown in Table 5.4.

Table 5.4: Verification of body-left wheel relation

Tables 5.4 and 5.5 present the results of the verification of the body-left wheel and body-right wheel relations, respectively, which proceed analogously to process of verifying the body-top relation just described. As shown in Table 5.5 (marked with an arrow), at t=38 a unit becomes active representing the successful recognition of a train.

Table 5.5: Verification of body-right wheel relation

Table 5.4: Verification of body-left wheel relation

| | t=23 | t=24 | t=25 | t=26 | t=27 | t=28 | t=29 | t=30 |
|---|---|---|---|---|---|---|---|---|
| trainckL.wheel | **9.0f** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| traincL.wheel | 0.0 | **9.0f** | **9.0f** | **9.0f** | **9.0f** | **9.0f** | **9.0f** | 0.2 |
| trainL.wheelOK | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **8.8f** |
| distance = 1 | 0.5 | **9.4f** | 6.1 | 4.0 | 2.6 | 1.7 | 1.1 | 0.7 |
| direction = LL. | 0.0 | **9.0f** | 5.9 | 3.8 | 2.5 | 1.6 | 1.0 | 0.7 |
| rotation = 60o | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** |
| dilation = 1 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| translation = (2,2) | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| 1from(2,2) | 5.0 | 5.0 | **9.7f** | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| 2from(2,2) | 5.0 | 5.0 | 7.2 | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| Lfrom(2,2) | 4.9 | 4.9 | 7.0 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| ULfrom(2,2) | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| URfrom(2,2) | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| Rfrom(2,2) | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LRfrom(2,2) | 4.9 | 4.9 | **9.4f** | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LLfrom(2,2) | 4.9 | 4.9 | 7.0 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| position(1,1) | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1,2) | 0.0 | 0.0 | 0.0 | **9.5f** | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1,3) | 0.0 | 0.0 | 0.0 | 2.4 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,1) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,3) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2,4) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,1) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,2) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3,3) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| lrg(1,2) | 0.0 | 0.0 | 0.0 | 0.0 | 4.8 | 3.1 | 2.0 | 1.3 |
| sml(1,2) | 4.0p | 4.0p | 4.0p | 4.0p | **8.8f** | 7.1p | 6.0p | 5.3p |
| red(1,2) | 0.0 | 0.0 | 0.0 | 0.0 | 4.8 | 3.1 | 2.0 | 1.3 |
| brn(1,2) | 4.0p | 4.0p | 4.0p | 4.0p | **8.8f** | 7.1p | 6.0p | 5.3p |
| dul(1,2) | 4.0p | 4.0p | 4.0p | 4.0p | **8.8f** | 7.1p | 6.0p | 5.3p |
| shn(1,2) | 0.0 | 0.0 | 0.0 | 0.0 | 4.8 | 3.1 | 2.0 | 1.3 |
| cyl(1,2) | 0.0 | 0.0 | 0.0 | 0.0 | 4.8 | 3.1 | 2.0 | 1.3 |
| sph(1,2) | 4.0p | 4.0p | 4.0p | 4.0p | **8.8f** | 7.1p | 6.0p | 5.3p |
| lrgsph | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.4 | 2.9 | 1.9 |
| smlsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | **8.9f** | 6.0 | 3.8 |
| redsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 4.5 | 2.9 | 1.9 |
| brnsph | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **8.8f** | 5.7 | 3.7 |
| dulsph | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **8.8f** | 5.7 | 3.7 |
| shnsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 4.5 | 2.9 | 1.9 |
| smlbrndulsph | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **8.8f** | 5.7 |

## Table 5.5: Verification of body-right wheel relation

| | t=31 | t=32 | t=33 | t=34 | t=35 | t=36 | t=37 | t=38 |
|---|---|---|---|---|---|---|---|---|
| trainckRwheel | **8.8f** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| trainctrain | 0.0 | **8.8f** | **8.8f** | **8.8f** | **8.8f** | **8.8f** | **8.8f** | 0.0 |
| trainOK | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **9.2f** ← |
| distance = 1 | 0.5 | **9.1f** | 5.9 | 3.8 | 2.5 | 1.6 | 1.1 | 0.7 |
| direction = LR | 0.0 | **8.8f** | 5.7 | 3.7 | 2.4 | 1.6 | 1.0 | 0.7 |
| rotation = 60o | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** | **9.5f** |
| dilation = 1 | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| translation = (2.2) | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** | **10.0f** |
| 1from(2.2) | 5.0 | 5.0 | **9.6f** | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| 2from(2.2) | 5.0 | 5.0 | 7.0 | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| Lfrom(2.2) | 4.9 | 4.9 | 7.0 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| ULfrom(2.2) | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| URfrom(2.2) | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| Rfrom(2.2) | 4.9 | 4.9 | **9.3f** | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LRfrom(2.2) | 4.9 | 4.9 | 6.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| LLfrom(2.2) | 4.9 | 4.9 | 6.9 | 4.9 | 4.9 | 4.9 | 4.9 | 4.9 |
| position(1.1) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1.2) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(1.3) | 0.0 | 0.0 | 0.0 | 2.3 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2.1) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2.2) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2.3) | 0.0 | 0.0 | 0.0 | **9.4f** | 0.0 | 0.0 | 0.0 | 0.0 |
| position(2.4) | 0.0 | 0.0 | 0.0 | 4.6 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3.1) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3.2) | 0.0 | 0.0 | 0.0 | 4.8 | 0.0 | 0.0 | 0.0 | 0.0 |
| position(3.3) | 0.0 | 0.0 | 0.0 | 2.3 | 0.0 | 0.0 | 0.0 | 0.0 |
| lrg(2.3) | 0.0 | 0.0 | 0.0 | 0.0 | 4.7 | 3.1 | 2.0 | 1.3 |
| sml(2.3) | 4.0p | 4.0p | 4.0p | 4.0p | **8.7f** | 7.0p | 6.0p | 5.3p |
| red(2.3) | 0.0 | 0.0 | 0.0 | 0.0 | 4.7 | 3.1 | 2.0 | 1.3 |
| brn(2.3) | 4.0p | 4.0p | 4.0p | 4.0p | **8.7f** | 7.0p | 6.0p | 5.3p |
| dul(2.3) | 4.0p | 4.0p | 4.0p | 4.0p | **8.7f** | 7.0p | 6.0p | 5.3p |
| shn(2.3) | 0.0 | 0.0 | 0.0 | 0.0 | 4.7 | 3.1 | 2.0 | 1.3 |
| cyl(2.3) | 0.0 | 0.0 | 0.0 | 0.0 | 4.7 | 3.1 | 2.0 | 1.3 |
| sph(2.3) | 4.0p | 4.0p | 4.0p | 4.0p | **8.7f** | 7.0p | 6.0p | 5.3p |
| lrgsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 4.5 | 2.9 | 1.9 |
| smlsph | 2.4 | 1.6 | 1.0 | 0.7 | 0.4 | **9.0f** | 5.8 | 3.8 |
| redsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 4.5 | 2.9 | 1.9 |
| brnsph | 2.4 | 1.6 | 1.0 | 0.7 | 0.4 | **9.0f** | 5.8 | 3.7 |
| dulsph | 2.4 | 1.6 | 1.0 | 0.7 | 0.4 | **9.0f** | 5.8 | 3.7 |
| shnsph | 1.2 | 0.8 | 0.5 | 0.3 | 0.2 | 4.5 | 2.9 | 1.9 |
| smlbrndulsph | 3.7 | 2.4 | 1.6 | 1.0 | 0.7 | 0.4 | **9.3f** | 6.0 |

## 6. Discussion and Conclusions

We have presented a connectionist model for recognizing objects using visual feature information. While the model was developed within the context of a drastically simplified visual domain, we claim that the general principles it embodies are useful in understanding the computations involved in real-world visual object recognition. We now discuss how the model relates to the relevant biological and behavioral data on this topic, as well as comment on other general characteristics of the model.

### 6.1 Plausibility

It is clear that at a realistic scale the model would require a massive number of units and connections. Given a visual field with approximately 1,000,000 positions, and the extremely large feature set required, one can appreciate the large number of feature-value and feature-pair units that would be required in a straightforward expansion of the model to realistic scale. The various combinations of all possible dilations, rotations and translations that the transform parameter units and binder units must encode would suggest that a comparable number of units are needed to construct the viewing transform mechanism as we have described it. What is perhaps worse, the model clearly violates the biological constraint of an upper limit of about 10,000 connections per unit. For example, translation parameter units would require millions of connections in a full-scale VTM.

Luckily there are many ways to drastically reduce the number of units and connections per unit that the model would require if our simple encoding scheme were used. Ballard and Sabbah [Ballard & Sabbah, 1982] specify how to reduce the unit requirements for the viewing transform by using split parameter spaces. In addition, the coarse and coarse-fine encoding techniques mentioned earlier (which employ units with varying sensitivity along a number of feature dimensions) can significantly reduce the number of units required to encode feature information. These and other techniques [Feldman, 1981; Hinton, 1981] appear sufficient to reduce the resource demands of the model to within the constraints imposed by biology.

An important point to notice is that while the resource demands of the SIM and VTM are extensive, these mechanisms are entirely knowledge-independent and can be hard-wired into the visual system. We assume that a large majority of strictly visual areas of cortex are dedicated to carrying out the processes within the RF and SFF and would include structures for the SIM and VTM.

A fair amount of behavioral data fits in well with the current model of object recognition. The fact that the process of identifying visual elements loses spatial information is in accord with the feature interference effects found by Treisman [Treisman, 1982] and Estes' theory of positional uncertainly [Estes, 1975]. Much evidence supports the notion that observers compute a viewing transform to map object frames to the viewer frame [Rock, 1973; Hinton, 1979; Jolicoeur & Kosslyn, 1983; Palmer, 1983] and mechanisms similar to our VTM have been developed previously [Hinton, 1981; Hrechanyk & Ballard, 1982]. The verification of a subpart relation by selectively using feature information at a particular position is thought to be related to the covert attention process of [Posner, 1978] and [Treisman & Gelade, 1980]. Finally, the sequentiality of the process for verifying different subpart relations is supported by data on the eye movements of observers during object perception [Antes, 1973] and mental transformation [Just & Carpenter, 1976].

The major hypothesis of the model is that the relations between subparts of an object are verified sequentially. As such, the model makes predictions on the speed and complexity of the verification process that are different than a model proposing simultaneous verification would make. In particular, the time to recognize an object with complex internal geometry should be proportional to the number of subpart relations it contains (as represented in our model). In addition, the knowledge-independence of the SIM and VTM has consequences for the kinds of processing tasks that should be resistent to top-down biases of context. In general, it should be possible to empirically test certain aspects of the model through psychological experimentation.

Other aspects of the model which fit in well with previous work are the hierarchical representation of the subparts and internal geometry of an object [Marr & Nishihara, 1978; Palmer, 1977] and the notion of a canonical form of object representation used in recognition [Palmer *et al.*, 1981]. In addition, the model was explicitly designed to be compatible with both lower-level visual feature extraction processes [Ballard, 1981] and higher-level knowledge representation and general inferential processes [Shastri & Feldman, 1984].

## 6.2 Limitations

While the model successfully deals with some difficult aspects of object recognition, there are a number of important visual phenomena that are not even addressed by the model as it stands. Issues such as occlusion, transparency, motion, multiple objects, and multiple instances of an object are difficult problems that a comprehensive recognition model must effectively address. [Feldman, 1984] describes some general approaches to these problems which are compatible with the current model, but no mechanisms have been specified in any detail.

A significant limitation of the current computational model is that it has been implemented and tested only for an extremely simple visual domain. It is difficult (but luckily not impossible) to express significant problems in object perception in a domain with only four features at each of ten positions. As a result, we are unable to address certain issues that would normally arise in a more complicated domain. One example is our exclusion of multi-leveled hierarchical object representations. While our model in principle can incorporate processes for handling such objects, our restricted domain does not allow feature information of a complex object to be represented in sufficient detail to address the issue.

While our decision to employ such a restricted visual domain was based primarily on limitations of the ISCON simulator, the notion of studying vision in the context of a simplified domain is important to our modeling approach. Given the massive amount of interacting information in vision, only by limiting the domain to manageable complexity can one hope to separate out the significance of each type of information and determine how it interacts with the rest of the system [c.f. Feldman, 1984]. The goal in designing a simplified domain is to include all of the information that is needed by a particular process in a form which allows it to be effectively manipulated and which generalizes to more complicated domains. To the extent that the SFF representation of our simplified domain accomplishes this goal, our model will generalize to, and be useful in understanding, more complex (and more realistic) visual domains.

Another rather severe limitation of the current model is that it does not include three-dimensional information in its representation of an object's internal geometry or in its transform mapping from this representation to positions in an image. Clearly, in real-world vision the three-dimensional structure of an object is included in its representation, and because of effects such as foreshortening this information must be used in the viewing transform.

Including three-dimensional information in the object representations in our domain is straightforward. Instead of simple distance and direction relations, spherical or cylindrical coordinates could be used to specify the spatial relation between the main subpart and the other subparts. Furthermore, except for the problems involved with occlusion, the three-dimensional viewing transform is only slightly more complicated that the two-dimensional case [Ballard & Sabbah, 1982]. Extending the VTM to map between three-dimensional reference frames should pose no great difficulties for our encoding schemes other than a large increase in the number of units and connections required.

The requirement that *all* subparts and relations be verified during object recognition is too stringent for a more realistic domain involving occlusion among multiple objects. A more plausible extension of the model would interpret the presence of each subpart and relation as providing *evidence* for the object, and various contextual factors would determine what amount of evidence is sufficient for recognition [c.f. Feldman & Shastri, 1984].

Finally, the plausibility of the recognition model we have presented depends on both higher- and lower-level processes that have yet to be integrated with it. Parameter networks must reliably maintain stable feature information in the SFF, and focus-of-attention mechansisms must restrict processing to a limited area of the SFF at any one time. A satisfactory recognition model would integrate all of these processes into a full model which would use image intensity values to determine what objects are present at what positions in the visual field. At this point, we must settle for the fact that our model and the higher- and lower-level processes on which it depends are, *in principle*, compatible.

### 6.3 Future Directions

The obvious next step in the development of the model is to extend its domain so that the limitations of the model due to the restrictiveness of the current domain can be overcome. This extension should allow multiple three-dimensional objects with multi-leveled hierarchical representations to be present in an image. The resulting development of the model would need to include processing between different levels in a complex object's representation (following [Hrechanyk & Ballard, 1982]), a three-dimensional viewing transform (following [Marr & Nishihara, 1978] and [Ballard & Sabbah, 1982]), and focus-of-attention mechanisms (following [Hrechanyk & Ballard, 1982] and [Feldman, 1984]). None of the problems associated with these extensions seem insurmountable. Beyond this, the model of visual object recognition that we have developed must be integrated within a more comprehensive specification of the operation and interaction of the RF, SFF, WKF and EF of the Four Frames model.

This work was supported by the Defense Advanced Research Projects Agency under grant No. N00014-82-K-0193.

# 7. Appendix

## 7.1 Computational Details

A *unit* is a computational entity consisting of

$\{q\}$ - a set of discrete *states*, $< 10$,
$p$ - a continuous value in $[0,10]$ called *potential*,
$v$ - an *output* value, integer in $[0,10]$,
$\underline{i}$ - a vector of *inputs*, $i_1, \ldots, i_n$

and relatively simple functions that update these values,

$p \leftarrow P(\underline{i},p,q)$ - potential function,
$q \leftarrow Q(\underline{i},p,q)$ - state function,
$v \leftarrow V(\underline{i},p,q)$ - output function.

The set of states used in a typical unit in our model is $\{$*resting, firing*$\}$. The form of the potential function often includes a *decay* constant $\delta$ which reduces the potential by a constant factor after each step in the computation. Because of the dynamic nature of the information processing in our model, we use a relatively high value of $\delta = 3.5$ in our implementation. The output function generally produces a non-zero result only if the potential exceeds a certain *threshold* level $\theta$. In our implementation, $\theta = 8.0$. The magnitude of a unit's output (when non-zero) is generally proportional to the unit's potential.

The standard potential, state and output functions used by most units in our recognition model are the following:

$$
\begin{aligned}
p_{t+1} &\leftarrow (p_t - r)(1 - \delta) + \max\{s_1, \ldots, s_m\} \\
q_{t+1} &\leftarrow \text{if } p_{t+1} > \theta \text{ then } \textit{firing} \text{ else } \textit{resting} \\
v_{t+1} &\leftarrow \text{if } p_{t+1} > \theta \text{ then } p_{t+1} \text{ else } 0.0
\end{aligned}
$$

where $s_i$ in the potential function is the sum of the weighted inputs to site i and r is the resting potential of the unit.

Feature-value units have an additional state of *primed* which is used to encode information of the past firing behavior of the unit. The state signifies that the unit was recently *firing* but is now no longer receiving sufficient input to exceed threshold. The potential function uses state information in such a way that it takes less input to activate a *primed* unit than a *resting* unit. Visual input is initially sufficient to activate these units from their *resting* state, from which they settle into a *primed* state. Top-down activation during either the main subpart parameter determination or selective activation from the VTM is sufficient to activate these *primed* units only. In this way the feature-value units can compute the necessary intersection of visual input and current activation.

Specifically, feature-value units employ a more complicated state function of

```
$q_{t+1}$  ←  if $p_{t+1} > \theta$ then firing
                 else if $q_t$=primed or $q_t$=firing then primed
                      else resting
```

to retain their previous firing history. In addition, feature-value units in the primed state have a resting potential of 4.0 (rather than 0.0).

The two most common variations in unit operation required by our model involve the role of decay in the potential function. Certain types of units, such as binder units, must be totally input driven. These units use a decay constant of 1.0 (full decay) in order to prevent the previous potential level from affecting the unit's behavior. Other types of units, such as transform parameter units, must remain active throughout the subpart verification process once they exceed threshold. These units use no decay ($\delta = 0.0$) in their potential function.

An *enable signal* is represented by a connection with weight 1.0 from a unit which has a constant output value of 10.0 whenever its potential exceeds threshold. The site functions of the unit to be enabled reduce the sum of the input values at each site by 10.0. In this way, the unit will become active only if an enable signal conjoins with the inputs at one of its sites.

In general, the weight on each of the n input connections at a site will be 1/n. For example, each link in a 3-way conjunctive connection would have weight 0.333. Weights on inhibitory links are reduced by one-half, and are negative.

The detailed site, potential, state and ouput functions and connections weights used in the implementation described in Section 5.5 are available from the author upon request.

## References

Antes, J.R., "The time course of picture viewing." *Journal of Experimental Psychology, 103*, 62-70, 1974.

Ballard, D.H. "Parameter networks: Towards a theory of low-level vision." *Artificial Intelligence, 22*, 235-267, 1984.

Ballard, D.H. and Brown, C.M. *Computer vision.* Englewood Cliffs, NJ: Prentice Hall, 1982.

Ballard, D.H. and Sabbah, D. "On shapes." TR92, Computer Science Dept., U. Rochester, February 1982.

Barlow, H.B. "Single units and sensation: A neuron doctrine for perceptual psychology?" *Perception, 1*, 371-394, 1972.

Barrow, H.G. and Tennenbaum, J.M. "Representation and use of knowledge in vision." Technical Note 108, AI Center, SRI International, Menlo Park, CA, 1975.

Barrow, H.G. and Tennenbaum, J.M. "Recovering intrinsic scene characteristics from images." In A.R. Hanson and E.M. Riseman (eds.) *Computer vision systems.* NY: Academic Press, 1978.

Estes, W.K. "The locus of inferential and perceptual processes in letter identification." *Journal of Experimental Psychology: General, 2*, 122-145, 1975.

Falk, G. "Interpretation of imperfect line data as a three-dimensional scene." *Artificial Intelligence, 4*, 101-144, 1972.

Feldman, J.A. "A connectionist model of visual memory." In G.E. Hinton and J.A. anderson (eds.) *Parallel models of associative memory.* Hillsdale, NJ: L. Erlbaum Associates, 1981.

Feldman, J.A. "Dynamic connections in neural networks." *Biological Cybernetics, 46*, 27-39, 1982.

Feldman, J.A. "Four frames suffice: A provisionary model of vision and space." TR99, Computer Science Dept., U. Rochester, to appear, *Behavioral and Brain Sciences*, 1984.

Feldman, J.A. and Ballard, D.H. "Connectionist models and their properties." *Cognitive Science, 6*, 205-254, 1982.

Feldman, J.A. and Shastri, L. "Evidential inference in activation networks." *Proceedings*, Cognitive Science Conference, Boulder, CO, June 1984.

Hinton, G.E. Draft of a Technical Report, U. California, San Diego, 1980.

Hinton, G.E. "A parallel computation that assigns canonical object-based frames of reference." *Proceedings*, 7th IJCAI, 683-685, Vancouver, B.C., August 1981.

Hrechanyk, L.M. and Ballard, D.H. "A connectionist model of form perception." *Proceedings*, IEEE Workshop on Computer Vision, 44-51, Rindge, NH, 1982.

Jolicoeur, P. "Abstract frames of reference." In preparation.

Jolicoeur, P. and Kosslyn, S.M. "Coordinate systems in the long-term memory representation of three-dimensional shapes." *Cognitive Psychology, 15*, 301-345, 1982.

Just, M.A. and Carpenter, P.A. "Eye fixations and cognitive processes." *Cognitive Psychology, 8*, 441-480, 1976.

Marr, D.C. *Vision*. San Francisco, CA: W.H. Freeman and Co., 1982.

Marr, D.C. and Nishihara, H.K. "Representation and recognition of the spatial organization of three-dimensional shapes." *Proceedings, Royal Society of London. Series B, 200*, 269-294, 1978.

Palmer, S.E. "Hierarchical structure in perceptual representation." *Cognitive Psychology, 9*, 441-474, 1977.

Palmer, S.E., Rosch, E. and Chase, P. "Canonical perspective and the perception of objects." In J. Long and A. Braddeley (eds.) *Attention and Performance IX*. Hillsdale, NJ: L. Erlbaum Associates, 1981.

Posner, M.I. *Chronometric explorations of mind.* Hillsdale, NJ: L. Erlbaum Associates, 1978.

Roberts, L.G. "Machine perception of three-dimensional solids." In J.T. Tippett et al. (eds.) *Optical and electro-optical information processing*, Cambridge, MA: M.I.T. Press, 1965.

Rock, I. *Orientation and form.* NY: Academic Press, 1973.

Small, S.L., Shastri, L., Brucks, M.L., Kaufman, S.G., Cottrell, G.W. and Addanki, S. "ISCON: A network construction aid and simulator for connectionist models." TR109, Computer Science Dept., U. Rochester, April, 1983.

Shastri, L. and Feldman, J.A. "Semantic networks and neural nets." TR131, Computer Science Dept., U. Rochester, January 1984.

Treisman, A.M. "The role of attention in object perception." *Preceedings*, The Royal Society International Symposium on Physical and Biological Processing of Images, London, September 1982.

Treisman, A.M. and Gelade, G. "A feature-integration theory of attention." *Cognitive Psychology, 12*, 97-136, 1980.

Waltz, D.I. "Generating semantic descriptions from drawings of scenes with shadows." In P.H. Winston (ed.) *The psychology of computer vision.* NY: McGraw-Hill, 1975.

Zucker, S., Rosenfeld, A. and Davis, L. "General purpose models: Expectations about the unexpected." *Proceedings*, 4th IJCAI, Tbilisi, Georgia, USSR, 1975.